8/1/72

DIGITGRAPH 72-019

digital

P. O Box 4249
Tucson, A. [illegible] 85717

# MAINDEC CHANGE NOTICE

30 MCN

CHANGE NO.
11-DZLAB-A
Sheet 1 of 1

| AUTHOR | PROGRAM DATE | PRODUCT LINE | MAINDEC NUMBER |
|---|---|---|---|
| J. Hittell<br>DATE EXT.<br>20 July '72 2173 | June 1972 | PDP11<br>V96-06807 | 11-DZLAB-1 |

PROGRAM NAME LC11/LA30 Terminal Test

DEVICE LC11 Serial

| ITEM | |
|---|---|
| Ø | For information - Program has been released to Program Library. |

D.G/TGRAPH
P.O. Box 4249
Tucson, Arizona 85717

# IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | MAINDEC-11-DZLAB-A-D |
| PRODUCT NAME: | LC11/LA30 TERMINAL TESTS |
| DATE: | JUNE 1972 |
| MAINTAINER: | DIAGNOSTIC GROUP |
| AUTHOR: | L. FRIEDRICH |

COPYRIGHT © 1972
DIGITAL EQUIPMENT CORPORATION

NOTE: THIS PROGRAM IS A MODIFIED VERSION OF THE TELETYPE DIAGNOSTIC MAINDEC-11-D2AB

1. ABSTRACT

THE LA30 TERMINAL TESTS CONSISTS OF A PACKAGE OF TEST PROGRAMS DESIGNED TO TEST THE LC11 INPUT-OUTPUT LOGIC, THE LA30 PRINTER AND KEYBOARD. ALL TESTS ARE INCLUDED IN ONE OBJECT TAPE. THIS TEST CAN TEST THE LA30 WHEN INTERFACED BY EITHER A SINGLE (KL) OR DOUBLE BUFFERED (DL) CONTROLLER COMMUNICATING SERIAL TO 300 BAUD.

UNDER MONITOR LOAD IT WILL TEST THE CONSOLE DEVICE AND ALL CONTIGUOUS EXTRA DEVICES.

THIS TEST IS COMPATIBLE FOR ALL CONFIGURATIONS OF PDP-11. THE AVAILABLE TEST PROGRAMS ARE LISTED HERE IN NUMERICAL ORDER:

PRG0-COMBINED INPUT-OUTPUT LOGIC TESTS
PRG1-DISPLAY TEST
PRG2-KEYBOARD TEST
PRG3-PRINTER EXERCISER
PRG4-CLOCK ADJUSTMENT ROUTINE
PRG5-CLOCK ADJUSTMENT ROUTINE
PRG6-MAINTENANCE MODE SINGLE CHARACTER DATA TEST.
PRG7-MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN TEST.
PRG10-NON-PRINTING CHARACTER TEST
PRG11-WORST CASE NOISE
PRG12-LAST CHARACTER VISIBILITY

2. REQUIREMENTS

2.1 EQUIPMENT

A. PDP-11. SYSTEM. (4 K CORE).
B. LC11 INTERFACE/LA30 TERMINAL
C. HIGH SPEED READER

THE LC11 MUST HAVE STANDARD TELETYPE ADDRESSES. REFER TO SECTION 7.2 IF THE LC11 DOES NOT HAVE STANDARD PERIPHERAL ADDRESSES.

2.2 STORAGE

THIS PROGRAM USES LOCATION 00200 THROUGH 14554.

3. LOADING PROCEDURE

```
THIS PROGRAM'S OBJECT TAPE IS PUNCHED IN ABSOLUTE FORMAT.
THE ABS LOADER IS USED TO LOAD THE PROGRAM.
```

4, USE PROCEDURE

4,1 PRG0 USE PROCEDURE

A, CHECK POWER SWITCH IS "ON"
B, LOAD ADDRESS 000200
C, SET SR TO 000000, PRESS START
D, THE PROGRAM STOPS AT COMMON HALT.
E, SET ANY DESIRED SR OPTIONS, NORMAL RUN IS WITH SR = 000000,

THIS PROGRAM'S SR OPTIONS ARE:

| | |
|---|---|
| SR15 | HALT AT END OF ROUTINE |
| SR14 | ENTER SCOPE MODE AFTER ERROR |
| SR11 | INHIBIT ITERATION |
| SR10 | LOOP PROGRAM |
| SR9 | SELECT ROUTINE |

SR6 THROUGH SR0 = NUMBER OF ROUTINE TO BE SELECTED,

SECTION 7,1 GIVES A COMPLETE EXPLANATION OF SR OPTIONS,

F, PRESS CONTINUE, THE PROGRAM IS EXECUTED AND STOPS AT PROGRAM END HALT WHEN COMPLETED, PROVIDED NO ERRORS OCCUR.
G, REFER TO SECTION 6, IF ERRORS OCCUR.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8,1

EXECUTION TIME:

ONE NORMAL ERROR FREE PASS TAKES APPROXIMATELY 4 MINUTES,

4.2 PRG1 USE PROCEDURE

A. SET LA30 TO ON
B. LOAD ADDRESS 000200.
C. SET SR TO 000001. PRESS START
D. PROGRAM STOPS AT COMMON HALT.
E. SET ANY DESIRED SR OPTIONS. NORMAL RUN IS WITH SR = 000000.

THIS PROGRAM'S SR OPTIONS ARE:

SR15 HALT AT END OF ROUTINE
SR10 LOOP PROGRAM
SR9 SELECT ROUTINE
SR8 THROUGH SR0 = NUMBER OF ROUTINE TO BE SELECTED.

SECTION 7.2 GIVES A COMPLETE EXPLANATION OF SR OPTIONS.

F. PRESS CONTINUE. THE LA30 WILL BE EXERCISED AND
THE PROGRAM WILL STOP AT PROGRAM END HALT WHEN COMPLETED.
G. ERROR DETECTION IS BY VISUAL INSPECTION OF DISPLAY.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.3

EXECUTION TIME:

ONE NORMAL PASS TAKES APPROXIMATELY 12 MINUTES.

4.3 PRG2 USE PROCEDURE

A. SET LA30 ON-LINE.
B. LOAD ADDRESS 000200.
C. SET SR TO 000002. PRESS START
D. THE PROGRAM TYPES "KEYBOARD TEST" AND STOPS AT COMMON HALT.
E. SET ANY DESIRED SR OPTIONS. NORMAL RUN IS WITH SR = 000000.
THIS PROGRAM'S SR OPTIONS ARE:

SR15 HALT AT END OF ROUTINE
SR10 LOOP PROGRAM
SR9 SELECT ROUTINE
SR6 THROUGH SR0 = NUMBER OF ROUTINE TO BE SELECTED.

SECTION 7.1 GIVES A COMPLETE EXPLANATION OF SR OPTIONS.

F. PRESS CONTINUE. FOLLOW TYPED INSTRUCTIONS. WHEN DONE
PROGRAM STOPS AT PROGRAM END HALT.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.5

EXECUTION TIME:

PROGRAM IS USER DEPENDENT.

4.4 PRG3 USE PROCEDURE

A. SET LA30 TO ON
B. LOAD ADDRESS 000200
C. SET SR TO 000003, PRESS START
D. THE PROGRAM TYPES "TYPE IN DATA"
E. KEY IN ANY FIVE CHARACTERS TO BE TYPED.
F. KEY IN EITHER A RUBOUT FOR FULL SPEED TYPING, OR ANY OTHER CHARACTER FOR RANDOM STALLS BETWEEN CHARACTERS.
G. THE PROGRAM TYPES CONTINUOUSLY LINES CONTAINING THE FIVE CHARACTERS SPECIFIED, UNTIL SR15 IS SET TO A 1, AT THAT POINT THE PROGRAM GOES TO STEP E.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.8

## 4.5 PRG4 USE PROCEDURE

PRG4 IS USED AS AN AID IN ADJUSTING THE TRANSMITTER CLOCK, AND IN OBSERVING THE DATA BITS AS THEY ARE SHIFTED OUT OF THE TRANSMITTER BUFFER. A SCOPE IS REQUIRED.

TO ADJUST THE PUNCH CLOCK PROCEED AS FOLLOWS:

A. LOAD ADDRESS 000200
B. SET SR TO 00004, PRESS START.
C. PROGRAM STOPS AT COMMON HALT.
D. SET ANY DESIRED ASCII CODE IN LEFT HALF OF SR.
E. SET NUMBER OF MILLISECONDS TO DELAY BETWEEN PUNCH COMMANDS IN RIGHT HALF OF SR. THE NUMBER OF MILLISECONDS SELECTED SHOULD BE LONG ENOUGH FOR THE ENTIRE PUNCH OPERATION TO COMPLETE. A SUGGESTED STARTING NUMBER IS 177.
F. PRESS CONTINUE. THE PROGRAM RUNS CONTINUOUSLY. FIRST IT LOADS THE PUNCH BUFFER WITH THE CHARACTER IN SR LEFT, AND THEN DELAYS FOR THE NUMBER OF MILLISECONDS SPECIFIED IN SR RIGHT BEFORE RELOADING THE PUNCH BUFFER AGAIN.
G. SET UP A SCOPE AND DISPLAY THE PUNCH CLOCK PULSES. ADJUST THE PUNCH CLOCK ACCORDING TO SPECIFICATIONS.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 6.10

## 4.6 PRG5 USE PROCEDURE

PRG5 IS USED AS AN AID IN ADJUSTING THE RECEIVER CLOCK, AND IN OBSERVING THE DATA BITS AS THEY ARE SHIFTED INTO THE RECEIVER BUFFER. A SCOPE IS REQUIRED.

THE PROGRAM MAKES USE OF THE TRANSMIT MAINTENANCE BIT FEATURE IN ORDER TO CAUSE THE DATA OUTPUTTED TO THE TRANSMITTER BUFFER TO BE SHIFTED INTO THE RECEIVER BUFFER.

TO ADJUST THE RECEIVER CLOCK PROCEED AS FOLLOWS:

A. LOAD ADDRESS 000200
B. SET SR TO 000005, PRESS START.
C. PROGRAM STOPS AT COMMON HALT.
D. SET ANY DESIRED ASCII CODE IN LEFT HALF OF SR.
E. SET NUMBER OF MILLISECONDS TO DELAY BETWEEN TRANSMIT COMMANDS IN RIGHT HALF OF SR. THE SELECTED NUMBER SHOULD BE LONG ENOUGH FOR THE ENTIRE TRANSMIT/RECEIVE OPERATION TO COMPLETE. A SUGGESTED STARTING NUMBER IS 177.
F. PRESS CONTINUE. THE PROGRAM RUNS CONTINUOUSLY. FIRST IT LOADS THE TRANSMITTER BUFFER WITH THE CHARACTER IN SR LEFT, AND THEN DELAYS THE NUMBER OF MILLISECONDS SPECIFIED IN SR RIGHT. AS THE DATA BITS ARE SHIFTED OUT OF THE TRANSMITTER BUFFER, THE RECEIVER CLOCK STARTS, AND THE DATA BITS ARE SHIFTED INTO THE RECEIVER BUFFER. AT THE END OF THE DELAY THE PROGRAM MOVES THE RECEIVER BUFFER CONTENTS TO REG 0, AND ISSUES 5 RESET INSTRUCTIONS IN ORDER TO MAKE THE RECEIVER BUFFER CONTENTS VISIBLE IN THE RIGHT HALF OF THE DATA LIGHTS.
G. SET UP A SCOPE AND DISPLAY THE RECEIVER CLOCK PULSES. ADJUST THE RECEIVER CLOCK ACCORDING TO SPECIFICATIONS.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.11

4,7 PRG6 USE PROCEDURE

A, LOAD ADRESS 000200',
B', SET SR TO 000006', PRESS START
C', THE PROGRAM STOPS AT COMMON HALT,
D', SET CODE FOR CHARACTER TO BE TESTED IN THE LEFT HALF OF THE SR',
E, PRESS CONTINUE', THE PROGRAM RUNS CONTINUOUSLY, OUTPUTING THE CHARACTER TO THE OUTPUT BUFFER AND CHECKING THAT THE RECEIVE BUFFER CONTAINS THE SAME CHARACTER WHEN THE RECEIVE DONE BIT BECOMES SET,
F', REFER TO SECTION 6', ERRORS, IF ERRORS OCCUR',
G', THIS TEST CANNOT BE USED WITH A PARALLEL LA30 INTERFACE BECAUSE OF THE DIFFERENCE IN THE MAINTENANCE FEATURE',

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8',12

EXECUTION TIME!

CONTINUOUSLY RUNNING PROGRAM',

4,8 PRG7 USE PROCEDURE

A, LOAD ADDRESS 000200',
B', SET SR TO 00007', PRESS START
C', THE PROGRAM RUNS CONTINUOUSLY', THE SPECIAL BINARY COUNT PATERN IS OUTPUTTED TO THE OUTPUT BUFFER', EACH TIME THE RECEIVE DONE BIT BECOMES SET THE CHARACTER IN THE RECEIVE BUFFER IS CHECKED TO SEE THAT IT MATCHES THE PREVIOUSLY OUTPUTTED CHARACTER', THE PROGRAM STALLS RANDOMLY BETWEEN CHARACTERS', TO RUN AT FULL SPEED, SET SR8 TO A 1',
D', REFER TO SECTION 6', ERRORS, IF ERRORS OCCUR',
E', THIS TEST CANNOT BE USED WITH A PARALLEL LA30 INTERFACE BECAUSE OF THE DIFFERENCE IN THE MAINTENANCE FEATURE',

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8',13

EXECUTION TIME!

CONTINUOUSLY RUNNING PROGRAM',

```
4,9     PRG10 USER PROCEDURE
        ------------------

        A,  LOAD ADDRESS 200',
        B,  SET SR TO 000010', PRESS START',
        C,  PROGRAM RUNS CONTINUOUSLY', AN A IS PRINTED FOLLOWED BY
            AN ATTEMPT TO PRINT ALL NON-PRINTING CHARACTERS, THIS
            SEQUENCE IS REPEATED FOR A FULL LINE OF A'S',
        D,  REFER TO SECTION 6', ERRORS, IF ERRORS OCCUR',

        EXECUTION TIME!

        CONTINUOUSLY RUNNING PROGRAM',

4,10    PRG11 USER PROCEDURE
        ------------------

        A,  LOAD ADDRESS 000200',
        B,  SET SR TO 000011', PRESS START',
        C,  THIS PROGRAM WAS WRITTEN TO SERVE AS A WORST CASE NOISE
            TEST IF A WORST CASE PATTERN IS EVER ESTABLISHED', THE
            PROGRAM ACCEPTS UP TO 80 CHARACTERS FROM THE KEYBOARD
            OR UNTIL A RUBOUT IS DETECTED', AT THAT POINT THE PROGRAM
            CONTINUOUSLY ECHOS THE CHARACTER STRING', SWITCH 15
            SELECTS NEW CHARACTER STRING',
        D,  REFER TO SECTION 6', ERRORS, IF ERRORS OCCUR',

        EXECUTION TIME!

        PROGRAM RUNS CONTINUOUSLY',

4,11    PRG12 USER PROCEDURE
        ------------------

        A,  LOAD ADDRESS 000200',
        B,  SET SR TO 000012', PRESS START',
        C,  THIS TEST VERIFIES THE "LAST CHARACTER" VISIBILITY FEATURE
            OF THE LA30 PRINT HEAD BY WAITING TWO SECONDS BETWEEN
            CHARACTERS',
        D,  REFER TO SECTION 6', ERRORS, IF ERRORS ARE DETECTED',

        EXECUTION TIME!

        PROGRAM RUNS CONTINUOUSLY',
```

4,12 MONITOR LOAD USE PROCEDURE - ACT11 OR DDP1 - NON PAPER TAPE SYSTEM

A, SET SR TO 000000,
B, WHEN LOADED BY THE MONITOR THIS TEST WILL AUTOMATICALLY EXECUTE PROG 0 & 1 FOR THE CONSOLE DEVICE AND FOR ALL CONTIGUOUSLY ASSIGNED EXTRA DEVICES, IN ADDITION PROGRAM 0 RUNS PROGRAM 7 AS A SUBROUTINE TO VERIFY THE CORRECT TRANSFER OF DATA THRU THE INTERFACE
C, REFER TO SECTION 6, ERRORS, ERRORS DETECTED ARE LOGGED ON THE CONSOLE DEVICE,

EXECUTION TIME!

APPROXIMATELY 20 MIN, PER DEVICE FOR ERROR FREE PASS,

4,13 GROUP TEST USE PROCEDURE

A, LOAD CONSTANTS AS DESCRIBED IN 7,3,
B, LOAD ADDRESS 000042 TO 000200,
C, SET SR TO 000000, PRESS START,
D, THIS TEST WILL AUTOMATICALLY EXECUTE PROG 0 & 1 FOR THE CONSOLE DEVICE AND FOR THE GROUP OF EXTRA DEVICES AS DEFINED IN 7,3, IN ADDITION PROGRAM 0 RUNS PROGRAM 7 AS A SUBROUTINE TO VERIFY THE CORRECT TRANSFER OF DATA THRU THE INTERFACE,
E, REFER TO SECTION 6, ERRORS, ERRORS DETECTED ARE LOGGED ON THE CONSOLE DEVICE,

EXECUTION TIME!

PROGRAM RUNS CONTINUOUSLY,

## 5; PROGRAM AND/OR OPERATOR ACTION

### 5.1 NORMAL HALTS

LOC CHLTA — COMMON HALT, THIS HALT OCCURS WHENEVER THE PROGRAM IS AWAITING USER INTERVENTION; THE DATA LIGHTS CONTAIN THE ADDRESS OF INSTRUCTION THAT GENERATED THE CALL TO THE COMMON HALT.

LOC RHLT — END OF ROUTINE HALT, THIS HALT OCCURS AT THE END OF A TEST ROUTINE IF SR18 IS SET TO A 1, TO PROCEED; PRESS CONTINUE; PROGRAMS PRG0, PRG1, AND PRG2 USE THE ROUTINE END OPTION;

LOC PRGEND — PROGRAM END HALT, THIS HALT NORMALLY OCCURS AT THE END OF PROGRAMS PRG0, PRG1 AND UNLESS THE LOOP PROGRAM OPTION IS SET; (SR18) THIS HALT WILL NOT OCCUR AT THE END WHEN THE OF PROG 0 & 1 PROGRAM HAS BEEN LOADED BY A MONITOR OR IF THE CONSTANTS PER 7,3 HAVE BEEN ALTERED.

## 6. ERRORS

### 6.1 ERROR HALTS

LOC ERRH — UNCONDITIONAL ERROR HALT. DATA LIGHTS CONTAIN ADDRESS OF INSTRUCTION THAT GENERATED THE ERROR CALL. REFER TO PROGRAM LISTING.

LOC CERRH — CONDITIONAL ERROR HALT. THIS CALL WILL ALWAYS OCCUR, UNLESS SR14 IS SET TO A 1 (SCOPE MODE) AND THE ERROR HAS OCCURRED AT LEAST ONCE. DATA LIGHTS CONTAIN ADDRESS OF INSTRUCTION THAT GENERATED ERROR CALL. REFER TO PROGRAM LISTING.

LOC DERRH — DATA ERROR HALT. OCCURS WHEN A PROGRAM OR ROUTINE CHECKING DATA FINDS THAT THE EXPECTED AND THE RECEIVED DATA DO NOT AGREE. THE LEFT HALF OF THE DATA LIGHTS CONTAIN THE EXPECTED 9 BIT DATA. THE RIGHT HALF CONTAINS THE RECEIVED 9 BIT DATA.

UNDER MONITOR LOAD AND WHEN MANUALLY ADJUSTED PER 7.3 FOR GROUP TESTING, AN ERROR MESSAGE IS TYPED ON THE CONSOLE DEVICE INDICATING TYPE OF ERROR, ADDRESS OF DEVICE IN ERROR, AND PROGRAM ADDRESS OF TEST. UNDER THESE CONDITIONS NO HALT OCCURS.

WITH MONITOR LOAD OR MANUAL SELECTION OF GROUP TEST A FAILURE IN THE CONSOLE DEVICE OR IMPROPER SL SETTING WILL GENERATE A TEST ABORTED MESSAGE.

## 6.2 NON RECOVERABLE ERROR HALTS

A NON-RECOVERABLE ERROR HALT WILL OCCUR AT THE ADDRESSES LISTED BELOW IF THROUGH HARDWARE OR SOFTWARE FAILURE, PROGRAM CONTROL IS TRANSFERRED TO AN UNEXPECTED AREA BETWEEN 000000 AND 000176.

| | |
|---|---|
| 000002 | RESERVED AREA |
| 000006 | ERROR TRAP |
| 000012 | RESERVED INSTRUCTION TRAP |
| 000016 | DEBUG TRAP |
| 000022 | IOT TRAP |
| 000026 | POWER FAIL TRAP |
| 000044 THROUGH 000776 | SYSTEM SOFTWARE AND INTERRUPT VECTOR AREA, EXCEPT FOR KL11 INTERRUPT VECTORS. |

TO FIND OUT WHERE THE PROGRAM WAS AT THE TIME THE FAILURE OCCURRED, PERFORM THE FOLLOWING STEPS:

A. EXAMINE THE CONTENTS OF REGISTER 6 (ADDRESS 177706).
B. TRANSFER THE CONTENTS OF REGISTER 6 TO THE SR, LOAD ADDRESS, AND EXAMINE.
C. THE DATA SHOWN IN THE DATA LIGHTS IS THE VALUE OF THE PC WHEN THE FAILURE OCCURRED.
D. LOCATE IN THE PROGRAM LISTING THE DISPLAYED PC VALUE.
E. THE INSTRUCTION THAT IMMEDIATELY PRECEDES THE ONE REFERENCED BY THE DISPLAYED PC VALUE IS THE INSTRUCTION THAT WAS BEING EXECUTED WHEN THE FAILURE OCCURRED.

A NON-RECOVERABLE ERROR HALT FAILURE IS AN ABNORMAL CONDITION INDICATING A HARDWARE FAILURE, OR MOST UNLIKELY, A PROGRAM FAILURE. THIS PROGRAM ASSUMES THAT THE PROCESSOR IS IN OPERATING CONDITION IN ORDER TO TEST THE LA36. ANY FURTHER STEPS TO DIAGNOSE A NON-RECOVERABLE ERROR ARE NOT WITHIN THE SCOPE OF THIS PROGRAM.

## 7. MISCELLANEOUS

### 7.1 SR OPTIONS

THE STANDARD SR OPTIONS ARE DESCRIBED HERE.

SR15 - HALT AT END OF ROUTINE. FOR THESE PROGRAMS CONSISTING OF A SET OF SEPARATE TEST ROUTINES, SR15 SET TO A 1 CAUSES THE PROGRAM TO HALT UPON COMPLETION OF THE ROUTINE CURRENTLY BEING EXECUTED. THREE POSSIBLE USES OF THIS OPTION ARE:

A. TO STEP THROUGH A PROGRAM ONE ROUTINE AT A TIME.

B. WHEN AN UNPREDICTED FAILURE HAS OCCURRED (BLOW UP, HANG UP), TO ADVANCE THROUGH THE PROGRAM ONE ROUTINE AT A TIME UNTIL THE FAILURE OCCURS. THE ROUTINE FOLLOWING THE LAST IDENTIFIED ROUTINE WOULD BE THE FAILING ROUTINE.

C. WHEN A PROGRAM IS IN EXECUTION, TO DETERMINE HOW FAR THE PROGRAM HAS PROGRESSED.

SR14 - SCOPE. THIS OPTION IS USED ONLY BY PROG. THE OPTION CAUSES THE PROGRAM TO BYPASS ERROR HALTS, AND TO STAY IN THE FAILING ROUTINE. THIS OPTION WILL NOT BECOME ACTIVE UNTIL AN ERROR OCCURS. SR14 MUST BE ON BEFORE THE ERROR OCCURS, OR AT LEAST IT MUST BE SET BEFORE PRESSING CONTINUE AFTER AN ERROR HALT.

(7.1 CONT'D)

SR11 - INHIBIT ITERATION COUNT. THIS OPTION IS USED BY PRG0, PRG1, AND PRG3. THESE PROGRAMS CONSIST OF A SET OF ROUTINES EACH OF WHICH SPECIFIES THE NUMBER OF TIMES A TEST IS TO BE PERFORMED BY MEANS OF AN ITERATION COUNT. SETTING SR11 TO A 1 CAUSES THE PROGRAM TO DISREGARD THE ITERATION COUNT AND PERFORM THE TEST ONLY ONCE FOR EACH ROUTINE. TWO POSSIBLE USES OF THIS OPTION ARE:

A. QUICK PASS. THE USER MAY ELECT TO RUN THROUGH A PROGRAM QUICKLY TO FIND OUT IF ANY FAILURES SHOW IMMEDIATELY. A SUCCESSFUL QUICK PASS HOWEVER, DOES NOT GUARANTEE THAT THE SAME PROGRAM WILL RUN ERROR-FREE WHEN PERFORMING A NORMAL ITERATION PASS.

B. SKIP OVER FAILING ROUTINE. WHEN A ROUTINE HAS DETECTED A SOLID FAILURE, THE ERROR WILL BE REPORTED MANY TIMES. TO GO ON TO THE NEXT ROUTINE, THE USER CAN INHIBIT ITERATION. IT WILL BE NECESSARY TO CAUSE THE PROGRAM TO STOP AT THE END OF THE ROUTINE BY SETTING SR15 TO A 1. OTHERWISE THE PROGRAM WOULD QUICKLY RUN THROUGH THE NEXT ROUTINE(S) ALSO.

SR10 - LOOP PROGRAM. THIS OPTION IS USED BY PROGRAMS PRG0, PRG1, AND PRG4. SETTING SR10 TO A 1 CAUSES THE PROGRAM TO REPEAT ITSELF UPON COMPLETION, INSTEAD OF STOPPING AT PROGRAM END HALT.

SR9 - SELECT ROUTINE. THIS OPTION IS USED BY PROGRAMS PRG0, PRG1 AND PRG4. THE USER MAY ELECT TO RUN ONLY ONE SPECIFIC ROUTINE BY SETTING SR9 TO A 1, AND SR8 THROUGH SR0 TO THE NUMBER OF THE DESIRED ROUTINE. REFER TO THE INDIVIDUAL PROGRAM DESCRIPTION IN SECTION 8 TO OBTAIN THE ROUTINE NUMBER. THE ROUTINE NUMBER SELECTED MUST BE A VALID NUMBER, OR AN ERROR HALT WILL OCCUR. THE SELECT ROUTINE OPTION WILL BE HONORED BY THE PROGRAM UPON COMPLETION OF THE CURRENT ROUTINE, OR UPON STARTING THE PROGRAM.

SR8 - DISABLE STALL MODE AND RUN FULL SPEED. USED BY PROGRAM PRG10. THIS PROGRAM OPERATES NORMALLY IN STALL MODE (TESTS OR EXERCISES ARE NOT FULL SPEED, BUT RANDOM DURATION DELAYS ARE INTRODUCED). SETTING SR8 TO A 1 CAUSE THE PROGRAM TO PERFORM THEIR

TESTS AT FULL SPEED'

16

7.2 TESTING LC11 AT NON-STANDARD ADDRESSES AND/OR VECTORS

THIS PROGRAM CAN TEST A LCI1 ASSIGNED TO NON-STANDARD
ADDRESSES AND VECTORS PROVIDED THESE ADDRESSES ARE PROVIDED
TO THE PROGRAM AS FOLLOWS!

A', IMMEDIATELY AFTER LOADING THE PROGRAM CHANGE THE FOLLOWING
LOCATIONS', REFER TO PROGRAM LISTING',

| LOCATION | FROM STANDARD | TO NON-STANDARD |
|---|---|---|
| CONADD | 177560 | RECEIVER CSR ADDRESS |
| CONVEC | 000060 | RECEIVER VECTOR ADDRESS |

B', PROCEED TO USE PROGRAM',

7.3 TESTING A CONTIGUOUS GROUP OF LC11'S WHEN THE TEST HAS BEEN LOADED MANUALLY'

THIS PROGRAM WILL AUTOMATICALLY AND CONTINUOUSLY TEST THE
CONSOLE DEVICE AND A CONTIGUOUS GROUP OF LC11'S PROVIDED CER-
TAIN CONSTANTS OF THE PROGRAM ARE CHANGED AS FOLLOWS!

A', IMMEDIATLY AFTER LOADING THE PROGRAM CHANGE THE FOLLOWING
LOCATIONS', REFER TO THE PROGRAM LISTINGS',

| LOCATION | FROM STANDARD | TO NON-STANDARD |
|---|---|---|
| 000042 | 000000 | 000200 |
| LOWADD | 176500 | TO FIRST ADDITIONAL LCI1 RECEIVER CSR ADDRESS |
| LSTADD | 000000 | TO LAST +10 LCI1 RECEIVER CSR ADDRESS |

B', RESET SR,
START AT 200',

8. DESCRIPTION

NOTE! THIS TEXT WAS ORIGINALLY WRITTEN TO DESCRIBE A TELETYPE; THEREFORE, INTERPRET READER/KYBD AS RECEIVER AND PUNCH AS TRANSMITTER;

8.1 PRG0 PROGRAM DESCRIPTION

PRG0 TESTS THE INPUT AND OUTPUT LOGIC IN ONE PROGRAM;

RTN0 TESTS ABILITY TO REFERENCE THE READER/KYBD STATUS WORD (TKS) WITHOUT TRAPPING;
RTN1 TESTS ABILITY TO REFERENCE THE READER/KYBD BUFFER (TKB) WITHOUT TRAPPING;
RTN2 TESTS ABILITY TO REFERENCE THE PRINTER/PUNCH STATUS WORD (TPS) WITHOUT TRAPPING;
RTN3 TESTS ABILITY TO REFERENCE THE PRINTER/PUNCH BUFFER (TPB) WITHOUT TRAPPING;
RTN4 TESTS ABILITY TO SET AND CLEAR THE READER/KYBD ID BIT;
RTN5 CHECKS THAT READER /KYBD ID BIT CAN BE CLEARED WITH RESET INSTRUCTION;
RTN11 CHECKS THAT READER DONE BIT SETS NO LATER THAN 200 MSECS AFTER READER ENABLE;
RTN12 TESTS THAT READER/KYBD DONE BIT CAN BE READ RELIABLY;
RTN13 CHECKS THAT RESET INSTRUCTION CLEARS THE READER DONE BIT;
RTN14 CHECKS THAT REFERENCING READER BUFFER CLEARS DONE BIT;
RTN17 TESTS THAT READ BUFFER CAN BE READ RELIABLY;
RTN20 CHECKS THAT READER DONE BIT IS ABLE TO CAUSE AN INTERRUPT; IF THE INTERRUPT IS SERVICED, IT WILL HAVE OCCURRED AT CORRECT VECTOR;
RTN21 TESTS THAT READER DONE DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR IS AT THE SAME PRIORITY AS THE READER'S INTERRUPT REQUEST LEVEL;
RTN22 TESTS THAT READER DONE CAUSES INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY ONE LEVEL LOWER THAN THE READER'S INTERRUPT REQUEST LEVEL;
RTN23 CHECKS THAT READER DONE DOES NOT REINTERRUPT AFTER RTI INSTRUCTION WHEN DONE BIT IS LEFT SET;
RTN24 TESTS ABILITY TO SET AND CLEAR THE PUNCH ID BIT;

(8,1 CONT'D)

RTN24 TESTS ABILITY TO SET AND CLEAR THE PUNCH ID BIT;
RTN25 CHECKS THAT PUNCH ID BIT CAN BE CLEARED WITH RESET INSTRUCTION;
RTN30 TESTS THAT RESET SETS THE PUNCH READY BIT, AND THAT THE READY BIT CAN BE READ RELIABLY;
RTN31 TESTS THAT PUNCH READY IS CLEARED BY LOADING THE PUNCH BUFFER;
RTN32 TESTS THAT BYTE LOADING PUNCH BUFFER+1 DOES NOT CLEAR THE PUNCH READY BIT;
RTN33 CHECKS THAT THE PUNCH BECOMES READY NO LATER THAN 200 MSECS AFTER BUFFER LOAD;
RTN34 CHECKS THAT PUNCH READY BIT CAN CAUSE INTERRUPT, IF THE INTERRUPT IS SERVICED, IT WILL HAVE OCCURRED AT THE CORRECT VECTOR
RTN35 TESTS THAT PUNCH READY DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY AS THE READER'S INTERRUPT REQUEST LEVEL;
RTN36 TESTS THAT PUNCH READY CAUSES AN INTERRUPT WHEN THE PROCESSOR IS AT PRIORITY ONE LEVEL LOWER THAN THE PUNCH INTERRUPT REQUEST LEVEL;
RTN40 CHECKS THAT PUNCH READY CAUSES AN INTERRUPT IMMEDIATELY UPON LOWERING PROCESSOR PRIORITY TO 0;
RTN41 CHECKS FOR CORRECT OPERATION OF WAIT INSTRUCTION, (REFER TO PROGRAM LISTING);
RTN42 TESTS THAT LOADING PUNCH BUFFER WITH MAINTENANCE BIT SET CAUSES READER DONE BIT TO SET NO LATER THAN 200 MSECS;
RTN43 IF MONITOR ADDRESS IN NONZERO (LOADED BY THE MONITOR OR GROUP TESTING DESIRED) THIS ROUTINE WILL LINK TO PROG. 7 WHICH VERIFIES THAT PROPER TRANSFERS CAN BE EXECUTED TO AND FROM THE BUFFERS

## 8.2 PRG1 PROGRAM DESCRIPTION

PRG1 EXERCISES THE PRINTER FUNCTIONS, VERIFICATION OF PRINTER OPERATION IS PERFORMED VISUALLY BY USER, THE PROGRAM CONTAINS 29 ROUTINES NUMBERED FROM 00 TO 34(8),

RTN0 CARRIAGE RETURN TEST', CHECKS THAT CARRIAGE CAN CORRECTLY RETURN TO PRINT POSITION 0 FROM EVERY OTHER PRINT POSITION; VISUAL DISPLAY IS A WEDGE',

RTN1 RIGHT MARGIN TEST', THIS ROUTINE VERIFIES THAT THE RIGHT MARGIN IS CORRECTLY SET FOR 80 PRINT POSITIONS', THE TEST TYPES 83 CHARACTERS, IF THE RIGHT MARGIN IS CORRECTLY SET, THE TYPED LINE SHOULD LOOK AS FOLLOWS;

RTN2 SPACE TEST', CHECKS ABILITY OF THE LA30 TO SPACE CORRECTLY; THE TEST FIRST PRINTS REVERSE SLASHES (\) IN ALTERNATE PRINT POSITIONS, AND THEN FROM PRINT POSITION 0 TO EACH PRINT POSITION AND PRINTS A SLASH (/)', THE TYPEOUT SHOULD LOOK AS FOLLOWS;

RTN3 LINE FEED TEST', TESTS FOR ABILITY TO CORRECTLY PERFORM A LINE FEED', A RANDOM STALL OCCURS BETWEEN EACH LINE FEED', A CORRECTLY PERFORMED TEST WILL APPEAR AS DIAGONAL LINE BETWEEN PRINT POSITION 0 AND PRINT POSITION 88',

RTN4 SPACE TEST', CHECKS ABILITY OF THE DISPLAY TO SPACE TO POSITIONS 1 TO 7', THE FIRST LINE TYPED MARKS THE TAB POSITIONS', SUBSEQUENT LINES INCREMENT BY ONE SPACE EACH', THE TYPEOUT LOOKS AS FOLLOWS;

(8,2 CONT'D)

```
RTN5     TYPES LINE OF CHARACTERS ABC
RTN6     TYPES LINE OF CHARACTERS DEF
RTN7     TYPES LINE OF CHARACTERS GHI
RTN10    TYPES LINE OF CHARACTERS JKL
RTN11    TYPES LINE OF CHARACTERS MNO
RTN12    TYPES LINE OF CHARACTERS PQR
RTN13    TYPES LINE OF CHARACTERS STU
RTN14    TYPES LINE OF CHARACTERS VWX
RTN15    TYPES LINE OF CHARACTERS YZ@
RTN16    TYPES LINE OF CHARACTERS 123
RTN17    TYPES LINE OF CHARACTERS 456
RTN20    TYPES LINE OF CHARACTERS 789
RTN21    TYPES LINE OF CHARACTERS !"#
RTN22    TYPES LINE OF CHARACTERS $%&
RTN23    TYPES LINE OF CHARACTERS '()
RTN24    TYPES LINE OF CHARACTERS *+,
RTN25    TYPES LINE OF CHARACTERS -.;/
RTN26    TYPES LINE OF CHARACTERS :;<
RTN27    TYPES LINE OF CHARACTERS =>?
RTN30    TYPES LINE OF CHARACTERS @[\
RTN31    TYPES LINE OF CHARACTERS ]^ AND LEFT ARROW
RTN32    TYPES 2 LINES OF ALL CHARACTERS, FIRST LINE IS TYPED AT
         FULL SPEED, SECOND LINE IS TYPED WITH RANDOM STALLS.
```

8,3 PRG2 PROGRAM DESCRIPTION

PRG2 IS USED TO TEST THE LA36 KEYBOARD, THE PROGRAM CONTAINS 3 ROUTINES NUMBERED FROM 00 TO 02.

RTN0 TESTS THAT LC11 CONTROL RESPONDS WHEN USER DEPRESSES A KEYBOARD KEY.

RTN1 ECHO TEST, THE TEST ECHOES ONTO THE TELEPRINTER THE CHARACTER RECEIVED FROM THE KEYBOARD, WHEN THE TEST SENSES A RUBOUT CHARACTER THE TEST IS ENDED, THE TEST ENABLES THE USER TO DETERMINE IF ALL PRINTABLE CODES CAN BE SUCCESSFULLY SENT TO THE LC11 CONTROL. THE FOLLOWING SECTIONS (8,3,1, 8,3,2) DESCRIBE HOW THIS ROUTINE SHOULD BE USED TO TEST THE SPECIAL CHARACTERS.

RTN2 OCTAL EQUIVALENT TEST, THE OCTAL EQUIVALENT OF ANY CHARACTER RECEIVED BY THE CONTROL IS TYPED, SENSING A RUBOUT ENDS THE TEST. THIS TEST ENABLES THE USER TO DETERMINE THAT ALL CODES INCLUDING NON-PRINTABLE CONTROL CODES ARE BEING CORRECTLY SENT TO THE LC11 CONTROL.

8,4 PRG3 PROGRAM DESCRIPTION

PRG3 IS A PRINTER EXERCISER DESIGNED AS AN AID IN MAKING LASO ADJUSTMENTS. THE PROGRAM PERMITS THE USER TO TYPE IN FIVE TEST CHARACTERS AND ONE FINAL CHARACTER THAT SIGNIFIES WHETHER FULL SPEED OR STALL OPERATION IS DESIRED. THE PROGRAM THEN TYPES LINES CONTAINING THE FIVE SELECTED CHARACTERS. WHEN THE USER WISHES TO CHANGE THE TEST CHARACTERS SR15 IS SET TO A 1. THE PROGRAM TERMINATES TYPING THE LINE BEFORE ACCEPTING NEW DATA.

8,5 PRG4 PROGRAM DESCRIPTION

PRG4 IS USED AS AN AID IN ADJUSTING THE TRANSMITTER CLOCK WITH THE AID OF A SCOPE. THE PROGRAM PERFORMS THE FOLLOWING SEQUENCE:

A. LOAD TRANSMITTER BUFFER WITH ASCII CODE IN SR LEFT.
B. DELAY NUMBER OF MILLISECONDS SET IN SR RIGHT.
C. GO TO STEP A.

8,6 PRG5 PROGRAM DESCRIPTION

PRG5 IS USED AS AN AID IN ADJUSTING THE RECEIVER CLOCK. A SCOPE IS REQUIRED. THE PROGRAM PERFORMS THE FOLLOWING SEQUENCE:

A. SET PUNCH MAINTENANCE BIT.
B. LOAD PUNCH BUFFER WITH CODE IN SR LEFT.
C. DELAY NUMBER OF MILLISECONDS SET IN SR RIGHT.
D. MOVE CONTENTS OF READ BUFFER TO REGISTER B.
E. ISSUE 9 RESET INSTRUCTIONS TO "FIX" READ BUFFER CONTENTS IN RIGHT HALF OF DATA LIGHTS.
F. GO TO STEP A.

8,7 PRG6 PROGRAM DESCRIPTION

USING THE PUNCH MAINTENANCE BIT FEATURE, PRG6 TAKES THE ASCII CODE SET IN SR LEFT AND USES IT TO CHECK THE ABILITY OF THE CONTROL TO OUTPUT AND RECEIVE DATA. THE PROGRAM PERFORMS THE FOLLOWING SEQUENCE:

A. SET PUNCH MAINTENANCE BIT.
B. LOAD PUNCH BUFFER WITH CODE IN SR LEFT.
C. WHEN READER DONE BIT SETS, COMPARE CODE IN SR LEFT WITH DATA IN READER BUFFER. HALT IF NOT SAME.
D. WAIT FOR PUNCH DONE BIT TO SET AND GO TO STEP B.

8,8 PRG7 PROGRAM DESCRIPTION

USING THE PUNCH MAINTENANCE BIT FEATURE PRG7 USES THE SPECIAL BINARY COUNT PATTERN TO CHECK ABILITY OF THE CONTROL TO OUTPUT AND RECEIVE DATA. THE PROGRAM PERFORMS THE FOLLOWING STEPS:

A. INITIALIZE BINARY COUNT PATTERN;
B. SET PUNCH MAINTENANCE BIT;
C. LOAD PUNCH BUFFER WITH BINARY COUNT CHARACTER;
D. WHEN READER DONE BIT SETS, COMPARE BINARY CHARACTER WITH DATA IN READ BUFFER; HALT IF NOT SAME;
E. WAIT FOR PUNCH DONE BIT TO SET AND GO TO STEP C;
F. THIS PROGRAM IS EXECUTED AS A SUBROUTINE OF PROGRAM 0 UNDER MONITOR LOAD OR WHEN AUTOMATIC GROUP TESTING IS USED.

8,9 PROGRAM 10 DESCRIPTION

SEE SECTION 4.11 USER PROCEDURE

8,10 PROGRAM 11

SEE SECTION 4.12 USER PROCEDURE

8.11 PROGRAM 12 DESCRIPTION

SEE SECTION 4.13 USER PROCEDURE

.END

```
                        ;KL11  TESTS FOR THE LA30 TERMINAL
                        ;COPYRIGHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                        ;
                        ;PRG0= COMBINED INPUT-OUTPUT LOGIC TESTS.
                        ;PRG1= DISPLAY TEST;
                        ;PRG2= KEYBOARD TEST.
                        ;PRG3= PRINTER EXERCISER;
                        ;PRG4= PUNCH CLOCK ADJUSTMENT ROUTINE;
                        ;PRG5= READER CLOCK ADJUSTMENT ROUTINE;
                        ;PRG6= MAINTENANCE MODE SINGLE CHARACTER DATA TEST;
                        ;PRG7= MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN DATA TEST;
                        ;PRG10= NON-PRINTING CHARACTER TEST
                        ;PRG11= WORST CASE NOISE
                        ;PRG12= LAST CHARACTER VISIBILITY TEST

                        ;STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1 )
                        ;
                        ;SR 15 = HALT AT END OF ROUTINE;
                        ;SR 14 = SCOPE;
                        ;SR 11 = INHIBIT ITERATION;
                        ;SR 10 = LOOP PROGRAM;
                        ;SR  9 = SELECT ROUTINE;
                        ;SR 8 = DISABLE STALL MODE AND RUN FULL SPEED;
                        ;SR 6 THROUGH SR 0 = NUMBER OF ROUTINE TO BE SELECTED;

000000  000000                .=0
000000  000002                .+2                     ;UNASSIGNED TRAP
000002  000000                HALT
000004  000006        MACHER: .+2                     ;SP OVERFLOW, BUS ERROR TRAP
000006  000000                HALT
000010  000012                .+2                     ;RESERVED INSTRUCTION TRAP
000012  000000                HALT
000014  000016                .+2                     ;TRACE TRAP
000016  000000                HALT
000020  000022                .+2                     ;TRAP TO CALL IOX
000022  000000                HALT
000024  000026                .+2                     ;POWER FAIL TRAP
000026  000340                PRTY7
000030  000036                EMTINT                  ;EMT TRAP
000032  000340                PRTY7
000034  000102                TRPINT                  ;TRAP TRAP; SIMILAR TO EMT
000036  000340                PRTY7
000040  000042                .+2
000042  000000        MONITR: HALT                    ;TRAPPED FOR MONITOR LOAD
        000176                .REPT   176
                              .+2
                              HALT                    ;TRAPPED TO PREVIOUS ADDRESS;
                              .ENDR
000044  000046                .+2
000046  000000                HALT                    ;TRAPPED TO PREVIOUS ADDRESS;
000050  000052                .+2
000052  000000                HALT                    ;TRAPPED TO PREVIOUS ADDRESS;
000054  000056                .+2
000056  000000                HALT                    ;TRAPPED TO PREVIOUS ADDRESS;
000060  000062                .+2
```

```
000042  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000044  000046          .+2
000046  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000050  000052          .+2
000052  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000054  000056          .+2
000056  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000060  000062          .+2
000062  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000064  000066          .+2
000066  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000070  000072          .+2
000072  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000074  000076          .+2
000076  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000100  000102          .+2
000102  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000104  000106          .+2
000106  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000110  000112          .+2
000112  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000114  000116          .+2
000116  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000120  000122          .+2
000122  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000124  000126          .+2
000126  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000130  000132          .+2
000132  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000134  000136          .+2
000136  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000140  000142          .+2
000142  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000144  000146          .+2
000146  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000150  000152          .+2
000152  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000154  000156          .+2
000156  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000160  000162          .+2
000162  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000164  000166          .+2
000166  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000170  000172          .+2
000172  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000174  000176          .+2
000176  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000200  000202          .+2
000202  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000204  000206          .+2
000206  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000210  000212          .+2
000212  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000214  000216          .+2
000216  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000220  000222          .+2
000222  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000224  000226          .+2
000226  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000230  000232          .+2
000232  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
000234  000236          .+2
000236  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS',
```

```
000340  000342
000342  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000344  000346
000346  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000350  000352
000352  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000354  000356
000356  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000360  000362
000362  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000364  000366
000366  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000370  000372
000372  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000374  000376
000376  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000400  000402
000402  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000404  000406
000406  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000410  000412
000412  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000414  000416
000416  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000420  000422
000422  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000424  000426
000426  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000430  000432
000432  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000434  000436
000436  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000440  000442
000442  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000444  000446
000446  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000450  000452
000452  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000454  000456
000456  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000460  000462
000462  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000464  000466
000466  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000470  000472
000472  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000474  000476
000476  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000500  000502
000502  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000504  000506
000506  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000510  000512
000512  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
000514  000516
```

```
000416  000000
000420  000422
000422  000000
000424  000426
000426  000000
000430  000432
000432  000000
000434  000436
000436  000000
000440  000442
000442  000000
000444  000446
000446  000000
000450  000452
000452  000000
000454  000456
000456  000000
000460  000462
000462  000000
000464  000466
000466  000000
000470  000472
000472  000000
000474  000476
000476  000000
000500  000502
000502  000000
000504  000506
000506  000000
000510  000512
000512  000000
000514  000516
000516  000000
000520  000522
000522  000000
000524  000526
000526  000000
000530  000532
000532  000000
000534  000536
000536  000000
000540  000542
000542  000000
000544  000546
000546  000000
000550  000552
000552  000000
000554  000556
000556  000000
000560  000562
000562  000000
000564  000566
000566  000000
000570  000572
000572  000000
```

```
HALT                    ;TRAPPED TO PREVIOUS ADDRESS';
.+2
```

```
000574  000576                  .+2
000576  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000600  000602                  .+2
000602  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000604  000606                  .+2
000606  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000610  000612                  .+2
000612  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000614  000616                  .+2
000616  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000620  000622                  .+2
000622  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000624  000626                  .+2
000626  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000630  000632                  .+2
000632  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000634  000636                  .+2
000636  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000640  000642                  .+2
000642  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000644  000646                  .+2
000646  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000650  000652                  .+2
000652  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000654  000656                  .+2
000656  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000660  000662                  .+2
000662  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000664  000666                  .+2
000666  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000670  000672                  .+2
000672  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000674  000676                  .+2
000676  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000700  000702                  .+2
000702  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000704  000706                  .+2
000706  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000710  000712                  .+2
000712  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000714  000716                  .+2
000716  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000720  000722                  .+2
000722  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000724  000726                  .+2
000726  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000730  000732                  .+2
000732  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000734  000736                  .+2
000736  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000740  000742                  .+2
000742  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000744  000746                  .+2
000746  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000750  000752                  .+2
```

```
000752  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000754  000756          .-2
000756  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000760  000762          .-2
000762  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000764  000766          .-2
000766  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000770  000772          .-2
000772  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
000774  000776          .-2
000776  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
001000  001002          .-2
001002  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
001004  001006          .-2
001006  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
001010  001012          .-2
001012  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
001014  001016          .-2
001016  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
001020  001022          .-2
001022  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
001024  001026          .-2
001026  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
001030  001032          .-2
001032  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS;
```

```
                        ;EQUATE STATEMENTS
          177570                SR=177570
          177776                CC=177776
          177776                PSW=177776
          000240                NOP=240
          000000                OPEN=0
          100000                HLTSW=BIT15             ;HALT SWITCH DEFINITION
          040000                SCOPSW=BIT14            ;SCOPE SWITCH DEFINITION
          020000                NPRTSW=BIT13            ;INHIBIT PRINT SWITCH DEFINITION
          010000                NTRCSW=BIT12            ;INHIBIT TRACE SWITCH DEFINITION
          004000                NITRSW=BIT11            ;INHIBIT ITERATION SWITCH DEFINITION
          002000                LPRGSW=BIT10            ;LOOP PROGRAM SWITCH DEFINITION
          001000                SRTSW=BIT9              ;SELECT ROUTINE SWITCH DEFINITION
          000400                BYPMAN=BIT8             ;BYPASS MANUAL INTERVENTION DEFINITION;
          100000                MANUAL=BIT15
          100000                BIT15=100000
          040000                BIT14=40000
          020000                BIT13=20000
          010000                BIT12=10000
          004000                BIT11=4000
          002000                BIT10=2000
          001000                BIT9=1000
          000400                BIT8=400
          000200                BIT7=200
          000100                BIT6=100
          000040                BIT5=40
          000020                BIT4=20
          000010                BIT3=10
          000004                BIT2=4
          000002                BIT1=2
          000000                BIT0=0
          000001          R1=%1
          000002          R2=%2
          005726                POPSP=5726              ;POP THE STACK, SAME AS TST (6)+
          022626                POPSP2=022626           ;POP STACK TWICE, SAME AS CMP (6)+,(6)+
          000340                PRTY7=340               ;PRIORITY LEVEL DEFINITIONS
          000300                PRTY6=300
          000240                PRTY5=240
          000200                PRTY4=200
          000140                PRTY3=140
          000100                PRTY2=100
          000040                PRTY1=40
          000000                PRTY0=0
          104000                TYPE=EMT+0
          104001                TYPES=EMT+1
          104002                STALL=EMT+2
          104003                ERROR=EMT+3
          104004                DATCHK=EMT+4
          104005                CHALT=EMT+5
          104006                STRDRV=EMT+6
          104007                STPCHV=EMT+7
          104010                CHALT=EMT+10
          104011                SRESET=EMT+11
          104012                CHAIN=EMT+12
```

```
                104014                  CK35=EMT+14
                104016                  TYPLN3=EMT+16
                104017                  DATHLT=EMT+17
                104020                  SAVREG=EMT+20
                104021                  RSTREG=EMT+21
                104022                  CHKASR=EMT+22
                104400                  DELAY=TRAP+0
                000007                  BELL=007
                014555                  BLOCKA=.END
                014557                  BLOCK1=BLOCKA+2
                014647                  BLOCKB=BLOCKA+112
                014700                  BLKBB=BLOCKA+123
                014671                  BLOCK2=BLOCKA+114
                014702                  BLK2=BLOCKA+125
                015051                  BLOCKC=BLOCKA+224
                015012                  BLKCC=BLOCKA+235
                000200                  .=200
000200  000167  001730                  JMP     START           ;GO TO START OF PROGRAM.
                001304          .=.+1100                        ;GET CODE OUT OF VECTOR AREA
001304  000000  SPBOT:  0                                       ;BOTTOM OF STACK
001306  177560  CONADD:  177560                                 ;CONSOLE CSR CSR POINTER
001310  000060  CONVEC:  60                                     ;CONSOLE INTERRUPT VECTOR
001312  177560  TKS:    177560                                  ;LSR CSR
001314  177562  TKB:    177562                                  ;LSR BUFFER
001316  177564  TPS:    177564                                  ;LSP CSR
001320  177566  TPB:    177566                                  ;LSP BUFFER
001322  000060  TKVTR:  60                                     ;LSR INTERRUPT VECTOR
001324  000200  TKLVL:  PRTY4                                  ;LSR PRIORITY LEVEL
001326  000064  TPVTR:  64                                     ;LSP INTERRUPT VECTOR
001330  000200  TPLVL:  PRTY4                                  ;LSP PRIORITY LEVEL
001332  000001  TTYTYP:  1                                     ;LA30 & KSR35
001334  000000  PRGNUM:  OPEN                                  ;CONTAINS CURRENT PROGRAM#
001336  000000  K$TART:  OPEN                                  ;CURRENT PROGRAM START ADDRESS,
001340  000000  CURTST:  OPEN                                  ;CONTAINS ADDR OF CURRENT TEST,
001342  000000  RTNNO:  OPEN                                  ;CONTAINS CURRENT TEST #,
001344  000000  NXTST:  OPEN                                  ;CONTAINS ADDR OF NEXT TEST,
001346  000000  ICTR:   OPEN                                  ;CONTAINS CURRENT ITERATION COUNT
001350  000000  SCOPTR:  OPEN                                 ;CONTAINS CURRENT SCOPE POINTER,
001352  000000  PRGIDI:  OPEN                                 ;CONTAINS PROGRAM INDICATORS
001354  006062  PRGTAB:  PRG0                                 ;PRG0 START ADDRESS
001356  010306          PRG1                                  ;PRG1 START ADDRESS
001360  011546          PRG2                                  ;PRG2 START ADDRESS
001362  012046          PRG3                                  ;PRG3 START ADDRESS
001364  012226          PRG4                                  ;PRG4 START ADDRESS
001366  012236          PRG5                                  ;PRG5 START ADDRESS
001370  012326          PRG6                                  ;PRG6 START ADDRESS
001372  012400          PRG7                                  ;PRG7 START ADDRESS
001374  012512          PRG10                                 ;NON PRINTING CHAR TEST
001376  012652          PRG11                                 ;WORST CASE NOISE
001400  013004          PRG12                                 ;LAST CHAR VISIBILITY
```

```
001402  004266          EMTTAB: TYP             ;POINTER TO TYPEOUT ROUTINE
001404  004504                  TYPS            ;POINTER TO CHAINED MESSAGES ROUTINE
001406  004602                  STAL            ;POINTER TO RANDOM STALL ROUTINE
001410  001732                  ERR             ;POINTER TO ERROR ROUTINE
001412  001634                  DTCHK
001414  001476                  CHLT            ;COMMON HALT
001416  004016                  STLSRV
001420  004046                  STLSPV
001422  001520                  EHLT            ;POINTER TO ERROR HALT ROUTINE;
001424  004076                  SRSETT
001426  002512                  CHAINN
001430  003234                  CHK33
001432  003256                  CHK35
001434  003246                  CHK330
001436  003700                  TYPL3
001440  001722                  DTHLT
001442  003610                  SAVRG
001444  003650                  RSTRG
001446  003266                  CKASR
001450  004536          TRPTAB: DLY
001452  000000          RCNT:   OPEN            ;CHARACTER COUNT
001454  000000          CRBUF:  OPEN            ;HOLDS ONE CHARACTER FROM READER;
001456  000000          CHR1:   OPEN
001460  000000          CHR2:   OPEN
001462  000000          CHR3:   OPEN
001464  000000          ERCTR:  OPEN
001466  000000          CTRA:   OPEN
001470  000000          CTRB:   OPEN
001472  000000          CTRC:   OPEN
001474  000000          CTRD:   OPEN
```

```
                                        ;COMMON HALT ROUTINE
001476  022767  000000  176336  CHLT:   CMP     #0,MONITR       ;MONITOR LOAD?
001504  001004                          BNE     CHLTR           ;YES = EXIT
001506  011600                          MOV     @X6,X0          ;DEVELOP ADDRESS OF CALLER
001510  162700  000002                  SUB     #2,X0
001514  000000                  CHLTA:  HALT                    ;HALT, ADDRESS OF CALL INSTRUCTION
001516  000002                  CHLTR:  RTI                     ;IN DATA LIGHTS
                                        ;UNCONDITIONAL ERROR HALT ROUTINE
001520  022767  000000  176314  EHLT:   CMP     #0,MONITR       ;MONITOR LOAD?
001526  001435                          BEQ     EHLTA           ;NO = HLT;
001530  104011                          SRESET
001532  004767  001554          EHLTB:  JSR     X7,CONIT        ;SET UP CONSOLE POINTERS
001536  016700  002036                  MOV     CURADD,X0       ;SET UP DEVICE NUMBER
001542  162700  000010                  SUB     #10,X0          ;OF DISPLAY IN ERROR;
001546  012767  013160  003566          MOV     #ERRORA,A1ST    ;SET DEVICE ADDRESS
001554  004767  000330                  JSR     X7,ACNVE        ;CONVERT OCTAL TO ASCII
001560  112777  000067  003554          MOVB    #67,@A1ST
001566  011600                          MOV     @X6,X0          ;TRANSFER PROGRAM ERROR
001570  162700  000002                  SUB     #2,X0           ;ADDRESS TO MESSAGE
001574  012767  013207  003540          MOV     #ERRORB,A1ST
001602  004767  000302                  JSR     X7,ACNVE
001606  104000                          TYPE                    ;TYPE ERROR MESSAGE
001610  013116                          ERRORM
001612  004767  001752                  JSR     X7,RSTART
001616  000167  000432                  JMP     NSTART
001622  011600                  EHLTA:  MOV     @X6,X0          ;DEVELOP ADDRESS OF CALLER
001624  162700  000002                  SUB     #2,X0
001630  000000                  ERRH:   HALT                    ;HALT ADDRESS OF ERROR CALL
001632  000002                          RTI                     ;IN DATA LIGHTS;
                                        ;DATA CHECK ROUTINE;
001634  126767  177614  177613  DTCHK:  CMPB    CRBUF,CRBUF+1   ;COMPARE EXPECTED AND RECEIVED
001642  001432                          BEQ     DTCHKA          ;CHARS; BRANCH IF SAME;
001644  022767  000000  176170          CMP     #0,MONITR       ;MONITOR LOAD?
001652  001423                          BEQ     DTHLT           ;NO GO TO NORMAL HALT
001654  004767  001432                  JSR     X7,CONIT        ;SET UP CONSOLE POINTERS
001660  016700  001714                  MOV     CURADD,X0       ;SET UP DEVICE NUMBER OF
001664  162700  000010                  SUB     #10,X0          ;DISPLAY IN ERROR
001670  012767  013322  003444          MOV     #DERRA,A1ST     ;SET ERROR MESSAGE ADDRESS
001676  004767  000206                  JSR     X7,ACNVE        ;CONVERT OCTAL TO ASCII
001702  112777  000067  003432          MOVB    #67,@A1ST       ;MOST SIG;DIGIT = 7
001710  104000                          TYPE                    ;TYPE ERROR MESSAGE
001712  013275                          DERR
001714  004767  001450                  JSR     X7,RSTART
001720  000565                          BR      NSTART
001722  016700  177526          DTHLT:  MOV     CRBUF,X0        ;MOVE S/B AND WAS CHARS TO R0;
001726  000000                  DERRH:  HALT                    ;DATA ERROR HALT; GOOD CHAR IN
                                                                ;DATA LIGHTS LEFT; BAD CHAR IN DATA
001730  000002                  DTCHKA: RTI                     ;LIGHTS RIGHT, EXIT;
                                        ;CONDITIONAL ERROR HALT;
001732  032767  040000  175630  ERR:    BIT     #SCOPSW,SR      ;CHECK SCOPE SWITCH;
001740  001404                          BEQ     ERRA            ;BRANCH IF NO SCOPE DESIRED;
001742  005767  177404                  TST     PRGID           ;SCOPING WANTED; FIRST ERROR?
001746  100001                          BPL     ERRA            ;NO SCOPE IF FIRST ERROR;
001750  000002                          RTI                     ;SCOPE EXIT;
```

```
001752  052767  100000  177372  ERRA:   BIS     #BIT15,PRGID    ;SET ERROR INDICATOR;
001760  022767  000000  176054          CMP     #0,MONITR       ;MONITOR LOAD?
001766  001434                          BEQ     ERRB            ;NO = GO TO NORMAL HALT
001770  104011                          .RESET
001772  004767  001314                  JSR     %7,CONIT        ;SET UP CONSOLE POINTERS
001776  016700  001576                  MOV     CURADD,%0       ;SET UP DEVICE NUMBER
002002  162700  000010                  SUB     #10,%0          ;OP DISPLAY IN ERROR
002006  012767  013237  003326          MOV     #CERRA,A1ST     ;SET ERROR MESSAGE ADDRESS
002014  004767  000070                  JSR     %7,ACNVE        ;CONVERT OCTAL TO ASCII
002020  112777  000067  003314          MOVB    #67,@A1ST       ;MOST SIG DIGIT = 7
002026  012767  013266  003306          MOV     #CERRB,A1ST     ;TRANSFER PROGRAM ERROR
002034  011600                          MOV     @%6,%0          ;ADDRESS TO MESSAGE
002036  162700  000002                  SUB     #2,%0
002042  004767  000042                  JSR     %7,ACNVE        ;CONVERT OCTAL TO ASCII
002046  104000                          TYPE                    ;TYPE ERROR MESSAGE
002050  013217                          CERR
002052  004767  001312                  JSR     %7,RSTART
002056  000506                          BR      NSTART
002060  011600                  ERRB:   MOV     @%6,%0          ;DEVELOP ADDRESS OF
002062  162700  000002                  SUB     #2,%0           ;CALLER
002066  000000                  CERRH:  HALT                    ;ERROR HALT
002070  000002                          RTI                     ;EXIT
                                ;ROUTINE END HALT SUBROUTINE;
002072  005767  175472          SHALT:  TST     SR              ;CHECK HALT SWITCH;
002076  100003                          BPL     SHLTA           ;BRANCH IF NO HALT DESIRED;
002100  116700  177236                  MOVB    RTNNO,%0        ;CURRENT TEST # TO R0;
002104  000000                  RHLT:   HALT                    ;ROUTINE END HALT;
002106  000207                  SHLTA:  RTS     %7              ;EXIT;
002110  016701  003226          ACNVE:  MOV     A1ST,%1         ;TRANSFER ADDRESS TO R1
002114  062701  000006                  ADD     #6,%1           ;ADD 6 TO REF LSD
002120  012702  000006                  MOV     #6,%2           ;SET ROUTINE CTR TO 6
002124  010067  003228                  MOV     %0,ACNVX        ;TRANSFER NUMBER TO CONVERT ROUTINE REG
002130  000167  003232                  JMP     ACNVM           ;GO TO CONVERT ROUTINE
```

```
                                ;ROUTINE FOR MONITOR LOAD
002134  012706  001304          START:  MOV     #SPBOT,%6       ;SET BOTTOM OF STACK
002140  012767  003126  175656          MOV     #PFAIL,24       ;SET POWER FAIL POINTER
002146  016767  001430  001424          MOV     LOWADD,CURADD   ;INITIALIZE POINTER FOR START OF MULTIPLES
002154  012767  002212  175622          MOV     #END,4          ;INITIALIZE TIME OUT TRAP
002162  012767  000300  001416          MOV     #300,CURVEC     ;INITIALIZE VECTOR POINTER
002170  012700  174000                  MOV     #174000,%0      ;ADDRESS OF FIRST DC11 TO R0
002174  005710                  B1      TST     @%0             ;IS THIS DEVICE EQUPPED?
002176  062767  000010  001402          ADD     #10,CURVEC      ;NO ADD #10 TO VECTOR AND
002204  062700  000010                  ADD     #10,%0          ;ADDRESS
002210  000771                          BR      B
002212  022626                  END:    POPSP2
002214  016702  001342                  MOV     LOWADD,%2
002220  162702  176500                  SUB     #176500,%2      ;SUBTRACT GROUP OFFSET
002224  060267  001336                  ADD     %2,CURVEC       ;ADD GROUP BIAS TO VECTOR POINTER,
002230  004767  001006                  JSR     %7,CONIT        ;RESTORE POINTERS & VECTORS TO CONSOLE
002234  012767  000000  001334          MOV     #0,STNUM        ;RESET PROGRAM LOADER
002242  022767  000000  175572          CMP     #0,MONITR       ;IS THIS A MONITOR LOAD?
002250  001427                          BEQ     START0          ;NO - GO TO REGULAR START
002252  022767  000000  175310          CMP     #0,SR           ;ANY PANEL SW SET?
002260  001405                          BEQ     NSTART          ;NO - GO TO NEW START
002262  104000                          TYPE                    ;TYPE ABORT MESSAGE
002264  013331                          ABORT
002266  000005                          RESET
002270  000167  010566                  JMP     LOGICAL         ;RETURN TO MONITOR;
002274  012706  001304          NSTART: MOV     #SPBOT,%6       ;SET BOTTOM OF SP STACK
002300  005067  175472                  CLR     PSW
002304  012767  000006  175472          MOV     #6,MACHER
002312  005067  177024                  CLR     RTNND
002316  016700  001254                  MOV     STNUM,%0        ;TRANSFER PROGRAM NUMBER TO R0
002322  005267  001250                  INC     STNUM
002326  000402                          BR      CRTA
002330  012706  001304          START0: MOV     #SPBOT,%6       ;SET BOTTOM OF SP STACK;
002334  005067  175436                  CLR     PSW
002340  012767  000006  175436          MOV     #6,MACHER
002346  005067  176770                  CLR     RTNND
002352  016700  175212                  MOV     SR,%0           ;(SR) TO R0
002356  042700  177760                  BIC     #177760,%0      ;LIMIT (SR) TO BITS 3-0
002362  020027  000014                  CMP     %0,#14          ;COMPARE (SR) TO PROGRAM LIMIT
002366  101402                          BLOS    CRTA            ;VALID PROGRAM NUMBER?
002370  104010                  INCPRG: EHALT                   ;NO, INCORRECT PRG NUMBER
002372  000660                          BR      START           ;START OVER;
002374  005067  176752          CRTA:   CLR     PRGID
002400  010067  176730                  MOV     %0,PRGNUM       ;SAVE PROGRAM NUMBER AT PRGNUM
002404  006100                          ROL     %0              ;R0X2
002406  000170  001354                  JMP     @PRGTAB(0)      ;GO TO SELECTED PROGRAM;
002412  104005                  SRSET:  EHALT                   ;SET SR OPTIONS DESIRED
002414  016767  176716  176722  GETRDY: MOV     KSTART,NXTST    ;ADDR OF 1ST ROUTINE TO NXTST
002422  000167  000350                  JMP     CLEAN           ;GO CLEAN UP,
002426  004767  000240          GTRDYA: JSR     %7,FORWD        ;ROLL FORWARD TO "NEXT" ROUTINE;
002432  032767  001000  175130  GTRDYB: BIT     #SRTSW,SR       ;CHECK FOR SELECT ROUTINE SWITCH
002440  001003                          BNE     GTRDYC          ;BRANCH IF SELECT ROUTINE SWITCH IS SET,
002442  004767  000302                  JSR     %7,GOTST        ;GO RUN CURRENT ROUTINE;
002446  000460                          BR      CHNB            ;NO GO; MANUAL RTN BYPASSED;
```

```
002490  016700  175114          GTRDYC: MOV     $R,R0           ;(SR) TO R0
002494  042700  177000                  BIC     #177000,R0      ;MASK UNDESIRED BITS
002460  126700  176656                  CMPB    RTNNO,R0        ;COMPARE RTNNO TO (R0)
002464  001004                          BNE     GTRDYD          ;BRANCH IF ROUTINE NOT FOUND YET,
002466  004767  000256                  JSR     %7,GOTST        ;GO RUN ROUTINE;
002472  104010                          EMALT                   ;AND GO; MANUAL RTN SELECTED BYPASSED,
002474  000747                          BR      GETRDY
002476  022767  177777  176640  GTRDYD: CMP     #-1,NXTST       ;NO, CHECK FOR LAST ROUTINE;
002504  001350                          BNE     GTRDYA          ;LAST ROUTINE?
002506  104010                  INCRTN: EMALT                   ;YES, INCORRECT ROUTINE SELECTED;
002510  000741                          BR      GETRDY          ;START OVER,
002512  005767  176634          CHAINN: TST     PRGID           ;TEST ERROR BIT IN PRGID;
002516  100013                          BPL     CHNA            ;BRANCH IF ERROR BIT NOT SET;
002520  032767  040000  175042          BIT     #SCOPSW,SR      ;ERROR BIT SET; CHECK FOR SCOPE OPTION;
002526  001407                          BEQ     CHNA            ;SCOPE SWITCH SET IN SR?
002530  022767  177777  176612          CMP     #-1,SCOPTR      ;YES, CHECK SCOPE ENTRY POINTER
002536  001403                          BEQ     CHNA            ;BRANCH IF SCOPE ENTRY IS #1;
002540  017716  176604                  MOV     @SCOPTR,@%6     ;SET UP TO GO SCOPING
002544  000002                          RTI                     ;GO TO SCOPE ENTRY;
002546  042767  100000  176576  CHNA:   BIC     #BIT15,PRGID    ;CLEAR ERROR BIT IN PRGID;
002554  032767  004000  175006          BIT     #INTRSW,SR      ;TEST INHIBIT ITERATION SWITCH
002562  001004                          BNE     CHNAA           ;INHIBIT ITERATION?
002564  005367  176596                  DEC     ICTR            ;NO
002570  001401                          BEQ     CHNAA           ;COUNT 0?
002572  000002                          RTI                     ;NO, RETURN TO TEST ROUTINE
002574  022626          CHNAA:  POPSP2                          ;POP STACK TWICE
002576  104011                          SRESET
002600  004767  001104                  JSR     %7,RETPVT       ;RESTORE INTERRUPT VECTORS
002604  004767  177262                  JSR     %7,$HALT        ;GO HALT IF HALT SWITCH IS SET
```

```
002610  032767  001000  174752  CHNB1:  BIT     #SRTSW,SR       ;CHECK SELECT ROUTINE SWITCH
002616  001276                          BNE     GETRDY          ;SELECT ROUTINE SWITCH SET?
002620  022767  177777  176516          CMP     #-1,NXTST       ;NO,
002626  001275                          BNE     GTRDYA-4        ;LAST TEST?
002630  032767  002000  174732          BIT     #LPRGSW,SR      ;YES, TEST LOOP PROGRAM SWITCH,
002636  001266                          BNE     GETRDY          ;LOOP PROGRAM?
002640  022767  000000  175174          CMP     #0,MONITR       ;MONITOR LOAD?
002646  001407                          BEQ     PRGEND          ;YES - GO TO HALT
002650  026727  000722  000001          CMP     STNUM,#1        ;PROGRAM #1 COMPLETED?
002656  003006                          BLE     NSTART          ;YES GO TO NEW START
002660  004767  000504                  JSR     %7,RSTART       ;GO TO RESTART
002664  000603                          BR      NSTART
002666  000000          PRGEND: HALT                            ;NO, PROGRAM END,
002670  000747                          BR      CHNB
002672  016705  176446          FORWD:  MOV     NXTST,%5        ;ADDR OF NEXT ROUTINE TO R5,
002676  012567  176440                  MOV     (5)+,RTNNO      ;GET NEXT ROUTINE NUMBER,
002702  012567  176436                  MOV     (5)+,NXTST      ;GET ADDR OF NEXT "NEXT" ROUTINE,
002706  105767  176440                  TSTB    PRGID           ;CHECK IF PROGRAM SCOPE AND ! COUNT
002712  100407                          BMI     FORWDB          ;PARAMETERS, BRANCH IF NOT,
002714  012567  176426                  MOV     (5)+,ICTR       ;GET ITERATION COUNT,
002720  012567  176424                  MOV     (5)+,SCOPTR     ;GET SCOPE LOOP ENTRY POINTER,
002724  010567  176410          FORWDA: MOV     %5,CURTST       ;ADDR OF NOW CURRENT TEST TO CURTST,
002730  000207                          RTS     %7              ;EXIT FORWD SUBROUTINE,
002732  012767  177777  176410  FORWDB: MOV     #-1,SCOPTR      ;FORCE "NO SCOPE"
002740  012767  000001  176400          MOV     #1,ICTR         ;FORCE 1 COUNT OF 1
002746  000766                          BR      FORWDA
002750  005767  176366          GOTST:  TST     RTNNO           ;CHECK FOR MANUAL RTN,
002754  100005                          BPL     GOTSTA          ;BRANCH IF NOT MANUAL RTN,
002756  032767  000400  174604          BIT     #BYPMAN,SR      ;MANUAL RTN, BYPASS IT?
002764  001401                          BEQ     GOTSTA          ;NO, RUN IT,
002766  000207                          RTS     %7              ;BYPASS MANUAL ROUTINE,
002770  005726          GOTSTA: POPSP
002772  000177  176342                  JMP     @CURTST         ;GO RUN TEST,
002776  012767  001233  001156  CLEAN:  MOV     #1233,RP1       ;RESET RANDOM NOS,
003004  012767  005622  001152          MOV     #5622,RP2
003012  104011                          SRESET
003014  012767  000006  174762          MOV     #6,MACHER       ;RESET MACHER TRAP,
003022  005067  174760                  CLR     PSW
003026  012706  001304                  MOV     #SPBOT,%6       ;SET UP BOTTOM OF STACK,
003032  000167  177370                  JMP     GTRDYA
003036  011646          EMTINT: MOV     @%6,-(6)        ;GET SAVED PC,
003040  162716  000002                  SUB     #2,@%6          ;DECREMENT PC BY 2,
003044  017616  000000                  MOV     @(6),@%6
003050  121627  000022                  CMPB    @%6,#22         ;CHECK THAT CALL IS
003054  101402                          BLOS    EMTA            ;WITHIN LIMITS,
003056  000000                          HALT                    ;CALL NOT WITHIN LIMITS,
003060  000776                          BR      .-2
003062  006116          EMTA:   ROL     @%6             ;EMT ARG X 2,
003064  042716  177001                  BIC     #177701,@%6     ;REMOVE 7 MSB,
003070  062716  001402                  ADD     #EMTTAB,@%6     ;FORM EMT RTN ADDR,
003074  017616  000000                  MOV     @(6),@%6
003100  000136                          JMP     @(6)+           ;GO TO EMT ROUTINE,
003102  011646          TRPINT: MOV     @%6,-(6)        ;GET SAVED PC,
003104  162716  000002                  SUB     #2,@%6          ;DECREMENT PC BY 2,
```

```
003110  017616  000000          MOV     @(6),@X6
003114  121627  000000          CMPB    @X6,#0          ;CHECK THAT EMT
003120  101435                  BLOS    TRPA            ;IS WITHIN LIMITS,
003122  000000                  HALT                    ;TRAP CALL NOT IN LIMIT.
003124  000776                  BR      .-2

                                ;ENTER HERE FOR POWER FAIL

003126  010046          PFAIL:  MOV     %0,-(6)         ;SAVE REGISTER OR STACK
003130  010146                  MOV     %1,-(6)         ;WHEN POWERING DOWN
003132  010246                  MOV     %2,-(6)
003134  010346                  MOV     %3,-(6)
003136  010446                  MOV     %4,-(6)
003140  010546                  MOV     %5,-(6)
003142  016746  174656          MOV     24,-(6)
003146  010667  000010          MOV     %6,SAVR6        ;STORE STACK POSITION
003152  012767  003164  174644  MOV     #RESTART,24
003160  000000                  HALT                    ;HALT ON POWER DOWN NORMAL
003162  000000          SAVR6:  0                       ;STACK IS SAVED HERE
003164  016704  177772  RESTART:        MOV     SAVR6,%6        ;RESTORE REGISTER OFF STACK
003170  012667  174630          MOV     (6)+,24         ;WHEN POWERING UP
003174  012605                  MOV     (6)+,%5
003176  012604                  MOV     (6)+,%4
003200  012603                  MOV     (6)+,%3
003202  012602                  MOV     (6)+,%2
003204  012601                  MOV     (6)+,%1
003206  012600                  MOV     (6)+,%0
003210  000167  176720          JMP     START
```

```
003214  006116                  TRPA1:  ROL     @X6             ;TRAP ARG X 2,
003216  042716  177001                  BIC     #177001,@X6     ;REMOVE 7 MSB,
003222  062716  001450                  ADD     #TRPTAB,@X6     ;FORM TRAP RTN ADDR,
003226  017616  000000                  MOV     @(6),@X6
003232  000136                          JMP     @(6)+           ;GO TO TRAP ROUTINE,
003234  005767  176072          CHK33:  TST     TTYTYP          ;CHECK FOR 33,
003240  001002                          BNE     .+6             ;BRANCH IF NOT 33,
003242  062716  000002                  ADD     #2,@X6          ;+2 TO EXIT POINTER
003246  000002                  CHK330: RTI                     ;EXIT
003250  022767  000001  176054  CHK35:  CMP     #1,TTYTYP       ;CHECK FOR 35,
003256  001002                          BNE     .+6             ;BRANCH IF NOT 35,
003260  062716  000002                  ADD     #2,@X6          ;+2 TO EXIT POINTER
003264  000002                          RTI                     ;EXIT
003266  032767  000010  176036  CKASR:  BIT     #BIT3,TTYTYP    ;CHECK FOR ASR TTY,
003274  001001                          BNE     .+4             ;BRANCH IF NOT ASR,
003276  000002                          RTI                     ;ASR, EXIT;
003300  022626                          POPSP2                  ;POP STACK TWICE;
003302  012767  000001  176036          MOV     #1,ICTR         ;FORCE 1 COUNT TO A 1;
003310  104012                          CHAIN                   ;CHAIN TO BYPASS ROUTINE;
                        ;SUBROUTINE TO RESTORE POINTERS & VECTORS FOR CONSOLE OUTPUT
003312  016700  175770          CONIT:  MOV     CONADD,X0       ;CONSOLE ADDRESS TO R0
003316  010067  175770                  MOV     X0,TKS          ;SET LSR CSR TO CONSOLE
003322  005720                          TST     (0)+            ;ADD 2 TO R0
003324  010067  175764                  MOV     X0,TKB          ;SET LSR BUFFER TO CONSOLE
003330  005720                          TST     (0)+            ;ADD 2 TO R0
003332  010067  175760                  MOV     X0,TPS          ;SET LSP CSR TO CONSOLE
003336  005720                          TST     (0)+            ;ADD 2 TO R0
003340  010067  175754                  MOV     X0,TPB          ;SET LSP BUFFER TO CONSOLE
003344  016767  175720  175798          MOV     CONVEC,TKVTR    ;SET LSR INTERRUPT VECTOR TO CONSOLE
003352  016767  175732  175746          MOV     CONVEC,TPVTR    ;SET LSP BUFFER TO CONSOLE
003360  062767  000004  175748          ADD     #4,TPVTR
003366  000207                          RTS     X7;
                        ;SUBROUTINE FOR ADVANCING TEST;
003370  012767  000000  000200  RSTART: MOV     #0,STNUM        ;RESET PROGRAM LOADER
003376  022767  000000  174164          CMP     #0,SR           ;ANY PR SWITCH SET?
003404  001407                          BEQ     RSTA            ;NO - CONTINUE
003406  004767  177700                  JSR     X7,CONIT        ;RESTORE POINTER TO CONSOLE
003412  104000                          TYPE                    ;TYPE ABORT MESSAGE
003414  013331                          ABORT
003416  000005                          RESET
003420  000167  007436                  JMP     LOGICAL         ;EXIT TO MONITOR
003424  004767  000260          RSTA1:  JSR     X7,RETPVT       ;RESET INTERRUPT VECTORS
003430  016700  000144                  MOV     CURADD,X0       ;NEXT DEVICE ADD TO R0
003434  062767  000010  000136          ADD     #10,CURADD      ;COMPUTE ADDRESS FOR NEXT TEST;
003442  026767  000132  000134          CMP     CURADD,LSTADD   ;IS THIS LAST ADDRESS
003450  001437                          BEQ     RSTB            ;YES - GO TO EXIT
003452  010067  175634                  MOV     X0,TKS          ;NEW LSR CSR
003456  062700  000002                  ADD     #2,X0
003462  010067  175626                  MOV     X0,TKB          ;NEW LSR BUFFER
003466  062700  000002                  ADD     #2,X0
003472  010067  175620                  MOV     X0,TPS          ;NEW LSP CSR
003476  062700  000002                  ADD     #2,X0
003502  010067  175612                  MOV     X0,TPB          ;NEW LSP BUFFER
003506  016700  000074                  MOV     CURVEC,X0       ;NEXT VECTOR ADD TO R0
```

```
PALX11  V003    28-JUN-72       19:07   PAGE 38-1

003512  062767  000010  000066        ADD     #10,CURVEC      ;COMPUTE ADDRESS FOR NEXT VECTOR
003520  010067  175576                MOV     %0,TKVTR        ;NEW LSR INTERRUPT VECTOR
003524  062700  000004                ADD     #4,%0
003530  010067  175572                MOV     %0,TPVTR        ;NEW LSP INTERRUPT VECTOR
003534  012767  003550  174242        MOV     #RSTB,4         ;SETUP TRAP EXIT
003542  005777  175544                TST     @TKS            ;IS THIS DEVICE EQUIPPED?
003546  000412                        BR      RSTC            ;EXIT;
003550  016767  000024  000026  RSTB: MOV     CURADD,LSTADD   ;SETUP LAST ADDRESS
003556  004767  177530                JSR     %7,CONIT        ;RESTORE POINTER TO CONSOLE
003562  104000                        TYPE                    ;TYPE DONE MESSAGE
003564  013074                        DONE
003566  000005                        RESET
003570  000167  007266                JMP     LOGICAL         ;EXIT TO MONITOR
003574  000207          RSTC: RTS     %7
003576  000000          STNUM: OPEN
003600  000000          CURADD: OPEN
003602  176500          LOWADD: 176500
003604  000000          LSTADD: OPEN
003606  000000          CURVEC: OPEN
                        ;SAVE REGS 0 TO 4 SUBROUTINE;
003610  012667  000030  SAVRG: MOV     (6)+,SVRPC      ;SAVE PC AND PSW;
003614  012667  000026          MOV     (6)+,SVRPSW
003620  010446                  MOV     %4,-(6)         ;SAVE REGS 0 - 4
003622  010346                  MOV     %3,-(6)         ;IN STACK;
003624  010246                  MOV     %2,-(6)
003626  010146                  MOV     %1,-(6)
003630  010046                  MOV     %0,-(6)
003632  016746  000010          MOV     SVRPSW,-(6)     ;RESTORE PC AND PSW;
003636  016746  000002          MOV     SVRPC,-(6)
003642  000002                  RTI                     ;EXIT;
003644  000000          SVRPC: OPEN
003646  000000          SVRPSW: OPEN
                        ;RESTORE REGS 0 TO 4 SUBROUTINE;
003650  012667  000030  RSTRG: MOV     (6)+,RSTPC      ;SAVE PC AND PSW;
003654  012667  000026          MOV     (6)+,RSTPSW
003660  012600                  MOV     (6)+,%0         ;RESTORE REGS 0 - 4
003662  012601                  MOV     (6)+,%1         ;FROM STACK;
003664  012602                  MOV     (6)+,%2
003666  012603                  MOV     (6)+,%3
003670  012604                  MOV     (6)+,%4
003672  016746  000010          MOV     RSTPSW,-(6)     ;RESTORE PC AND PSW;
003676  016746  000002          MOV     RSTPC,-(6)
003702  000002                  RTI                     ;EXIT
003704  000000          RSTPC: OPEN
003706  000000          RSTPSW: OPEN
                        ;SUBROUTINE TO RESET INTERRUPT VECTORS
003710  016701  175406  RETPVT: MOV     TKVTR,%1        ;PLACE CURRENT TRAP VECTOR
003714  010100                  MOV     %1,%0           ;ADDRESS IN R1, AND R1 TO R0;
003716  062700  000002          ADD     #2,%0           ;ADD #2 TO R0
003722  010021                  MOV     %0,(1)+         ;STORE ;+2 AT TKVTR
003724  012721  000000          MOV     #0,(1)+         ;STORE HALT AT TKVTR+2
003730  062700  000004          ADD     #4,%0
003734  010021                  MOV     %0,(1)+         ;STORE ;+2 AT TPVTR
003736  012711  000000          MOV     #0,@%1          ;STORE HALT AT TPVTR+2
003742  012711  000000          MOV     #0,@%1
```

003746 000207 RTS %7

```
                        ;ROUTINE TO FETCH A CHARACTER
003750  012767  000310  000306  AREAD:  MOV     #200.,,BRCTR        ;SET UP DELAY COUNT;
003756  052777  000004  175332          BIS     #4,@TPS             ;SET MAINTENANCE BIT
003764  005077  175330                  CLR     @TPB                ;LOAD PUNCH BUFFER
003770  105777  175316          ARDA:   TSTB    @TKS                ;CHECK DONE BIT,
003774  100407                          BMI     ARDB                ;BRANCH IF DONE;
003776  104400                          DELAY                       ;DELAY 1 MILLISECOND;
004000  000001                          1
004002  005367  000256                  DEC     BRCTR               ;TIME UP?
004006  001370                          BNE     ARDA                ;BRANCH IF TIME NOT UP YET;
004010  104010                          EHALT                       ;ERROR; NO RESPONSE FROM READER,
004012  000756                          BR      AREAD               ;TRY AGAIN;
004014  000207          ARDB:   RTS     %7                          ;EXIT
                        ;ROUTINE TO SET LSR INTERRUPT VECTOR AND PRIORITY
004016  017667  000000  000012  STLSRV: MOV     @(6),STPRA+2        ;MOVE VECTOR ADDR TO STPRA+2
004024  062716  000002                  ADD     #2,@%6              ;SET UP EXIT
004030  016701  175266                  MOV     TKVTR,%1
004034  012721  000000          STPRA:  MOV     #OPEN,(1)+          ;SET VECTOR ADDRESS
004040  016721  175260                  MOV     TKLVL,(1)+          ;SET PRIORITY
004044  000002                          RTI                         ;EXIT
                        ;ROUTINE TO SET LSP INTERRUPT VECTOR AND PRIORITY;
004046  017667  000000  000012  STLSPV: MOV     @(6),STPPA+2        ;MOVE VECTOR ADDR TO STPPA+2
004054  062716  000002                  ADD     #2,@%6              ;SET UP EXIT
004060  016701  175242                  MOV     TPVTR,%1
004064  012721  000000          STPPA:  MOV     #OPEN,(1)+          ;SET VECTOR ADDRESS;
004070  016721  175234                  MOV     TPLVL,(1)+          ;SET PRIORITY
004074  000002                          RTI                         ;EXIT;
                        ;ROUTINE TO ISSUE RESET;
004076  012700  052525          SRSETT: MOV     #52525,%0                   ;DATA TO R0;
004102  005100                          COM     %0                  ;COMPLEMENT (R0);
004104  010067  177770                  MOV     %0,SRSETT+2         ;(R0) TO SRSETT+2;
004110  000005                          RESET                       ;;ISSUE RESET; (R0) IS
004112  000002                          RTI                         ;DISPLAYED; EXIT;
                        ;RANDOM NUMBER GENERATOR; ROUTINE EXITS WITH NUMBER IN REGISTER 0;
004114  016700  000042  RNGEN:  MOV     RP1,%0
004120  006100                          ROL     %0
004122  006100                          ROL     %0
004124  066700  000034                  ADD     RP2,%0
004130  010067  000026                  MOV     %0,RP1
004134  006100                          ROL     %0
004136  006100                          ROL     %0
004140  066700  000020                  ADD     RP2,%0
004144  006100                          ROL     %0
004146  006100                          ROL     %0
004150  010067  000010                  MOV     %0,RP2
004154  016700  000002                  MOV     RP1,%0
004160  000207                          RTS     %7                  ;EXIT; NUMBER IN R0
004162  001233          RP1:    1233
004164  005622          RP2:    5622
```

```
004166  104006                  BREAD:  STRDRV                  ;SET READER VECTOR
004170  004234                          BREADB                  ;TO BREADB
004172  052777  000101  175112          BIS     #101,@TKS       ;ENABLE LSR AND LSRI,
004200  012767  177777  000056          MOV     #177777,BRCTR   ;DELAY APPROX, 400 MSECS;
004206  005367  000052                  DEC     BRCTR
004212  001375                          BNE     .-4
004214  005077  175072                  CLR     @TKS            ;CLEAR LSRI ENABLE;
004220  104010                          EHALT                   ;NO RESPONSE HALT;
004222  000761                          BR      BREAD           ;TRY AGAIN;
004224  117767  175064  175222  BREADA: MOVB    @TKB,CRBUF      ;CHAR READ TO CRBUF;
004232  000207                          RTS     %7              ;EXIT SUBROUTINE;
004234  005077  175052          BREADB: CLR     @TKS            ;CLEAR LSR INTERRUPT ENABLE;
004240  105777  175046                  TSTB    @TKS            ;TEST FOR DONE;
004244  100003                          BPL     BREADC          ;BRANCH IF DONE NOT SET;
004246  012716  004224                  MOV     #BREADA,@X6     ;MODIFY INTERRUPT EXIT TO BREADA;
004252  000002                          RTI                     ;OK, EXIT INTERRUPT;
004254  000000                  BREADC: HALT                    ;HALT; DONE BIT NOT SET AFTER INTERRUPT,
004256  012716  004166                  MOV     #BREAD,@X6      ;SET UP TO RETRY;
004262  000002                          RTI                     ;EXIT INTERRUPT;
004264  000000                  BRCTR:  OPEN
                                ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER;
004266  011600                  TYP:    MOV     @X6,X0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS;
004270  062716  000002                  ADD     #2,@X6          ;SET UP EXIT;
004274  011000                          MOV     @X0,X0          ;ADDRESS OF MESSAGE TO R0;
004276  112067  000200          TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
004302  122767  000100  000172          CMPB    #100,TYPDAT     ;CHECK FOR"@"CHARACTER
004310  001013                          BNE     TYPC            ;BRANCH IF NOT"@";
004312  112767  000000  000162          MOVB    #0,TYPDAT       ;NULL TO CLEAR DOUBLE BUFFER
004320  004767  000046                  JSR     %7,OUTTYP
004324  112767  000000  000150          MOVB    #0,TYPDAT
004332  004767  000034                  JSR     %7,OUTTYP
004336  000002                          RTI                     ;TERMINATOR CHAR; DONE; EXIT;
004340  122767  000045  000134  TYPC:   CMPB    #45,TYPDAT      ;CHECK FOR"%";
004346  001442                          BEQ     TYPF            ;BRANCH IF"%"
004350  122767  000043  000124          CMPB    #43,TYPDAT      ;NOT"%";CHECK FOR"#";
004356  001443                          BEQ     TYPG            ;BRANCH IF "#"
004360  004767  000006                  JSR     %7,OUTTYP       ;TYPE CHAR IN TYPDAT
004364  000744                          BR      TYPA
                                ;SUBROUTINE TO INSERT NULLS AND PRINT CHARACTER;  USES REGISTER
                                ;LOCATION TYPDAT REFERENCED IN THE OUTPUT MESSAGE SUBROUTINE
                                ;AND LINKS TO TYPDAT REGISTER IN THE LSPCH SUBROUTINE;
004366  110067  000110          OUT:    MOVB    %0,TYPDAT       ;MOVE LSPCH CHAR TO TYPDAT;
004372  122767  000015  000102  OUTTYP: CMPB    #15,TYPDAT      ;CHECK FOR CARRIAGE RETURN
004400  001003                          BNE     OUTCAR          ;NO=GO TO OUTPUT
004402  012767  000011  000042  OUTNUL: MOV     #9.,NULL        ;SET NULL CTR TO 9;
004410  116777  000066  174702  OUTCAR: MOVB    TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
004416  105777  174674                  TSTB    @TPS            ;WAIT FOR DONE FLAG;
004422  100375                          BPL     .-4
004424  022767  000000  000020          CMP     #0,NULL         ;IS NULL CTR = 0
004432  001406                          BEQ     OUTEX           ;IF ZERO EXIT
004434  005367  000012                  DEC     NULL            ;DEC NULL CTR
004440  112767  000000  000034          MOVB    #0,TYPDAT       ;TRANSFER ZERO TO OUTPUT
004446  000760                          BR      OUTCAR
004450  000207                  OUTEX:  RTS     %7              ;EXIT
004452  000000                  NULL:   OPEN
```

```
MACXI1  V003    28-JUN-72       19:37   PAGE 34-1

004454  112767  000015  000020  TYPF:   MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
004462  004767  177704          JSR     %7,OUTTYP       ;GO TYPE CHAR;
004466  112767  000012  000006  TYPG:   MOVB    #12,TYPDAT      ;MOVE LF CODE TO TYPDAT;
004474  004767  177672          JSR     %7,OUTTYP       ;GO TYPE CHAR,
004500  000676                  BR      TYPA
004502  000000          TYPDAT: OPEN
                        ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
004504  011600          TYPS:   MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
004506  062716  000002          ADD     #2,@%6          ;UPDATE TO NEXT MESSAGE ADDRESS
004512  011067  000014          MOV     @%0,TYPSB       ;ADDRESS OF MESSAGE TO TYPSB
004516  022767  177777  000006  CMP     #-1,TYPSB       ;CHECK FOR TERMINATOR
004524  001001                  BNE     TYPSA           ;BRANCH IF NOT TERMINATOR;
004526  000002                  RTI                     ;TERMINATOR, EXIT
004530  104000          TYPSA:  TYPE                    ;CALL ON TYP SUB TO TYPE MESSAGE
004532  000000          TYPSB:  OPEN                    ;ADDRESS OF MESSAGE GOES HERE
004534  000763                  BR      TYPS            ;GO PROCESS NEXT MESSAGE
```

```
                                ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
004536  011667  000036          DLY:    MOV     @%6,DLCNT       ;GET DELAY COUNT ADDRESS,
004542  062716  000002                  ADD     #2,@%6          ;SET UP EXIT ADDRESS
004546  017746  000026                  MOV     @DLCNT,-(6)     ;DELAY COUNT TO STACK
004552  005067  173220                  CLR     PSW             ;SET PRIORITY 0
004556  012746  002260          DLYA:   MOV     #2260,-(6)      ;1 MSEC COUNT TO STACK
004562  005316                  DLYB:   DEC     @%6             ;DECREMENT 1 MSEC COUNT
004564  001376                          BNE     DLYB            ;BRANCH IF NOT 0,
004566  005726                          POPSP                   ;ZERO, UNCOVER MSECS, COUNT,
004570  005316                          DEC     @%6             ;DECREMENT IT
004572  001371                          BNE     DLYA            ;BR IF NOT DONE DELAYING
004574  005726                          POPSP                   ;DONE
004576  000002                          RTI                     ;EXIT,
004600  000000          DLCNT:  OPEN                            ;CONTAINS MILLISECONDS COUNT ADDRESS,

                                ;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS, MAXIMUM STALL
                                ;DETERMINED BY CONTENTS OF LOC STLMSK,
004602  032767  040000  174542  STAL:   BIT     #BIT14,PRGID    ;TEST FOR STALLS ALLOWED,
004610  001001                          BNE     STALAA          ;ALLOWED,
004612  000002                          RTI                     ;NOT ALLOWED,
004614  004767  177274          STALAA: JSR     %7,RNGEN        ;GO GET RANDOM NUMBER,
004620  046700  000014                  BIC     STLMSK,%0       ;# IN R0, APPLY STALL MASK,
004624  001404                          BEQ     STALB           ;BRANCH IF RESULT IS 0,
004626  010067  000002                  MOV     %0,STALA
004632  104400                          DELAY                   ;DELAY
004634  000000          STALA:  OPEN                            ;DELAY COUNT
004636  000002          STALB:  RTI                             ;DONE, EXIT,
004640  000000          STLMSK: OPEN                            ;STALL MASK,
                                ;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT
004642  004767  177246          GRCNT:  JSR     %7,RNGEN        ;GET RANDOM NUMBER
004646  046700  000010                  BIC     RCMSK,%0        ;APPLY MASK
004652  001773                          BEQ     GRCNT           ;TRY AGAIN IF RESULT 0
004654  010067  000004                  MOV     %0,RNCNT        ;COUNT TO RNCNT
004660  000207                          RTS     %7              ;EXIT,
004662  000000          RCMSK:  OPEN                            ;RANDOM CHARACTER MASK,
004664  000000          RNCNT:  OPEN                            ;RANDOM CHARACTER COUNT,
                                ;SUBROUTINE TO COMPARE DATA READ FROM READER AGAINST EXPECTED DATA AND REPORT ERRORS,
004666  004767  000260          BCHECK: JSR     %7,GTBIN        ;GET BIN CHARACTER(IN R0)
004672  110067  174597                  MOVB    %0,CRBUF+1      ;STB CHAR TO CRBUF+1
004676  126767  174592  174591          CMPB    CRBUF,CRBUF+1   ;COMPARE S/B AND WAS CHARS,
004704  001001                          BNE     .+4             ;BRANCH IF NOT SAME,
004706  000207                          RTS     %7              ;SAME, EXIT,
004710  104417                          DATHLT                  ;GO HALT AND DISPLAY DATA,
004712  005367  174546                  DEC     ERCTR           ;3 ERRORS?
004716  001002                          BNE     .+6             ;BRANCH IF NOT 3 YET,
004720  004767  000002                  JSR     %7,RSYNC        ;3 ERRORS, RESYNC READER,
004724  000207                          RTS     %7              ;EXIT,
```

```
                                        ;SUBROUTINE TO SYNC THE LSR TO A SPECIAL BINARY COUNT PATTERN TEST TAPE.
004726  004767  000162          BSYNC:  JSR     %7,INBIN        ;INITIALIZE BINARY PATTERN
004732  004767  177230                  JSR     %7,BREAD        ;READ CHAR AND STORE AT CHR1
004736  116767  174512  174512          MOVB    CRBUF,CHR1
004744  004767  177216                  JSR     %7,BREAD        ;READ CHAR AND STORE AT CHR2
004750  116767  174500  174502          MOVB    CRBUF,CHR2
004756  004767  177204                  JSR     %7,BREAD        ;READ CHAR AND STORE AT CHR3.
004762  116767  174466  174472          MOVB    CRBUF,CHR3
004770  004767  000012                  JSR     %7,SYNCA        ;GO SYNC
004774  000754                          BR      BSYNC           ;NO SYNC, TRY AGAIN.
004776  012767  000003  174460          MOV     #3,ERCTR
005004  000207                          RTS     %7              ;SUCCESS,EXIT.
005006  012767  001000  000074  SYNCA: MOV     #512.,SYCTRA    ;512 TO SYCTRA
005014  012767  000012  000070          MOV     #10.,SYCTRB     ;10 TO SYCTRB
005022  004767  000124          SYNCB: JSR     %7,GTBIN        ;GET BIN CHARACTER(CHAR IN R0)
005026  120067  174424                  CMPB    %0,CHR1         ;COMPARE TO CHR1
005032  001373                          BNE     SYNCB           ;BRANCH IF NOT EQUAL
005034  004767  000112                  JSR     %7,GTBIN        ;SAME, GET ANOTHER BIN CHAR.
005040  120067  174414                  CMPB    %0,CHR2         ;COMPARE TO CHR2
005044  001405                          BEQ     SYNCD           ;BRANCH IF EQUAL
005046  005367  000036                  DEC     SYCTRA          ;DECREMENT SYCTRA
005052  001363                          BNE     SYNCB           ;BRANCH IF NOT DONE 512 TIMES.
005054  104010          SYNCC:  EMALT                           ;DONE 512, SYNC ERROR.
005056  000207                          RTS     %7              ;ERROR EXIT
005060  004767  000046          SYNCD: JSR     %7,GTBIN        ;GET BIN CHARACTER
005064  120067  174372                  CMPB    %0,CHR3         ;COMPARE TO CHR3
005070  001404                          BEQ     SYNCE           ;BRANCH IF SAME
005072  005367  000014                  DEC     SYCTRB          ;DECREMENT SYCTRB.
005076  001351                          BNE     SYNCB           ;BRANCH IF NOT DONE 10 TIMES
005100  000765                          BR      SYNCC           ;SYNC ERROR, BRANCH
005102  062716  000002          SYNCE: ADD     #2,@%6          ;SET UP SUCCESS EXIT
005106  000207                          RTS     %7              ;EXIT.
005110  000000          SYCTRA: OPEN
005112  000000          SYCTRB: OPEN
                                        ;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
005114  012767  177777  000014  INBIN: MOV     #-1,RIND        ;SET ALL VARIABLES
005122  004567  000380                  JSR     %5,BMOVE        ;TO MINUS 1.
005126  005136                          RIND
005130  005137                          RIND+1
005132  000011                          11.
005134  000207                          RTS     %7              ;EXIT
005136  000000          RIND:   OPEN
005140  000000          PT0:    OPEN
005142  000000          PT1:    OPEN
005144  000000          PIND:   OPEN
005146  000000          PT0P:   OPEN
005150  000000          PT1P:   OPEN
```

```
                                        ;SPECIAL BINARY COUNT PATTERN SUBROUTINE, EXITS WITH BIN CHAR IN R0
005152  016767  177762  177762  GTBIN:  MOV     PT0,PT1         ;PREVIOUS BIN CHAR TO PT1
005160  005167  177756          COM     PT1
005164  005167  177746          COM     R1ND
005170  001002                  BNE     .+6
005172  005267  177744          INC     PT1
005176  042767  177400  177736  BIC     #177400,PT1     ;MASK TO 8 BITS
005204  016767  177732  177726  MOV     PT1,PT0         ;SAVE BIN CHAR IN PT0
005212  016700  177724          MOV     PT1,R0          ;BIN CHAR TO R0,
005216  000207                  RTS     R7              ;EXIT'
005220  016767  177722  177722  GTBINP: MOV     PT0P,PT1P       ;PREVIOUS BIN CHAR TO PT1P
005226  005167  177716          COM     PT1P
005232  005167  177706          COM     P1ND
005236  001002                  BNE     .+6
005240  005267  177704          INC     PT1P
005244  042767  177400  177676  BIC     #177400,PT1P    ;MASK TO 8 BITS,
005252  016767  177672  177666  MOV     PT1P,PT0P       ;SAVE BIN CHAR IN PT0P',
005260  016701  177664          MOV     PT1P,R1         ;BIN CHAR TO R1,
005264  000207                  RTS     R7              ;EXIT',
                                        ;OCTAL TO ASCII CONVERT ROUTINES
005266  012500                  ACNV6:  MOV     (R5)+,R0        ;CONVERT TO 6 ASCII', GET OCTAL ADDRESS
005270  012567  000012          MOV     (R5)+,ACNVB     ;GET ASCII ADDRESS
005274  004767  000052          JSR     R7,ACNV         ;CONVERT TO ASCII
005300  004567  000122          JSR     R5,BMOVE        ;MOVE 6 CHARS TO ASCII ADDRESS
005304  005342                  A1ST
005306  000000                  ACNVB:  OPEN
005310  000006                  6
005312  000205                  RTS     R5              ;EXIT
005314  012500                  ACNV4:  MOV     (R5)+,R0        ;CONVERT TO 4 ASCII', GET OCTAL ADDRESS
005316  012567  000012          MOV     (R5)+,ACNVC     ;GET ASCII ADDRESS
005322  004767  000024          JSR     R7,ACNV         ;CONVERT TO ASCII
005326  004567  000074          JSR     R5,BMOVE        ;MOVE 4 CHARS TO ASCII ADDRESS',
005332  005344                  A1ST+2
005334  000000                  ACNVC:  OPEN
005336  000004                  4
005340  000205                  RTS     R5              ;EXIT
005342  000000                  A1ST:   OPEN
005344  000000                  OPEN
005346  000000                  OPEN
005350  000000                  ACNVX:  OPEN
005352  012701  005350          ACNV:   MOV     #A1ST+6,R1      ;ADDR TO STORE ASCII TO R1
005356  012702  000006          MOV     #6,R2           ;6 TO R2
005362  011067  177762          MOV     @R5,ACNVX       ;OCTAL WORD TO ACNVX
005366  016703  177756          ACNVM:  MOV     ACNVX,R3
005372  042703  177770          BIC     #177770,R3      ;ISOLATE LEAST SIGNIFICANT OCTAL #
005376  062703  000060          ADD     #60,R3          ;ADD 60 TO CONVERT TO ASCII
005402  110341                  MOVB    R3,-(1)         ;STORE ASCII BYTE
005404  006067  177740          ROR     ACNVX           ;MOVE NEXT OCTAL DIGIT TO LEAST
005410  006067  177734          ROR     ACNVX           ;SIGNIFICANT POSITION
005414  006067  177730          ROR     ACNVX
005420  005302                  DEC     R2              ;DONE 6 TIMES?
005422  001361                  BNE     ACNVM           ;NO, REPEAT,
005424  000207                  RTS     R7              ;YES', EXIT',
```

```
                                ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES;
005426  104020                  BMOVE:  SAVREG                  ;SAVE REGS;
005430  012701                          MOV     (5)+,X1         ;GET "FROM" ADDRESS
005432  012702                          MOV     (5)+,X2         ;GET "TO" ADDRESS
005434  012503                          MOV     (5)+,X3         ;GET COUNT
005436  112122                  BMOVA:  MOVB    (1)+,(2)+       ;MOVE BYTE
005440  005303                          DEC     X3              ;DECREMENT COUNT
005442  001375                          BNE     BMOVA           ;BRANCH IF NOT DONE;
005444  104021                          RSTREG                  ;RESTORE REGS;
005446  000205                          RTS     X5              ;DONE EXIT
                                ;SUBROUTINE TO CHECK FOR PUNCH READY;
005450  105777  173642          CPRDY:  TSTB    @TPS            ;TEST FOR READY BIT;
005454  100001                          BPL     CPRDYA          ;BRANCH IF READY NOT SET;
005456  000207                          RTS     X7              ;OK, EXIT;
005460  104010                  CPRDYA: EHALT                   ;NOT READY, HALT;
005462  000772                          BR      CPRDY
                                ;SUBROUTINE TO PUNCH ON LSP CHARACTER IN REG 0;
005464  004767  177760          LSPCH:  JSR     X7,CPRDY        ;GO CHECK FOR PUNCH READY
005470  004767  176672                  JSR     X7,OUT          ;GO TO OUTPUT ROUTINE;
005474  005000                          CLR     X0
005476  000207                          RTS     X7              ;DONE; EXIT;
                                ;BINARY TO DECIMAL ASCII CONVERT SUBROUTINE;
005500  012700  014344          BDCNV:  MOV     #DECVAL,X0      ;SET UP ADDR TO STORE DECIMAL ASCII IN R0
005504  013501                          MOV     @(5)+,X1        ;BINARY VALUE TO R1;
005506  012702  005606                  MOV     #ADTENP,X2      ;ADDR OF TEN POWER STRING TO R2;
005512  012707  000005  000060          MOV     #5,CNVCTR       ;SET UP FOR 5 POWER CONVERSIONS;
005520  012267  000060          BDCNVA: MOV     (2)+,TENPWR     ;MOVE POWER OF TEN VALUE TO TENPWR;
005524  004767  000010                  JSR     X7,SUBTEN       ;PERFORM CONVERSION
005530  005367  000044                  DEC     CNVCTR          ;DONE 5 CONVERSIONS?
005534  001371                          BNE     BDCNVA          ;BRANCH IF NOT YET 5;
005536  000205                          RTS     X5              ;YES, EXIT;
005540  005067  000036          SUBTEN: CLR     DIGIT           ;CLEAR DIGIT
005544  166701  000034          SUBTNA: SUB     TENPWR,X1       ;SUBTRACT TEN POWER FROM BINARY VALUE;
005550  103403                          BCS     SUBTNB          ;BRANCH IF UNSUCCESSFUL SUBTRACTION;
005552  005267  000024                  INC     DIGIT
005556  000772                          BR      SUBTNA
005560  062701  000020          SUBTNB: ADD     TENPWR,X1       ;RESTORE SUBTRACTED VALUE;
005564  062767  000060  000010          ADD     #60,DIGIT       ;CONVERT (DIGIT) TO ASCII
005572  116700  000004                  MOVB    DIGIT,(0)+      ;MOVE ASCII CHAR TO DECVAL FIELD;
005576  000207                          RTS     X7              ;EXIT;
005600  000000          CNVCTR: OPEN
005602  000000          DIGIT:  OPEN
005604  000000          TENPWR: OPEN
005606  023420          ADTENP: 10000.;
005610  001750                  1000.;
005612  000144                  100.;
005614  000012                  10.;
005616  000001                  1
```

```
                                ;SUBROUTINE TO TYPE A LINE OF CHARACTERS
005620  012767  000122  000050  TYPLN:  MOV     #82.,,TCTR      ;80 TO CHAR COUNT +CR,LF
005626  012704  014555          TYPLA:  MOV     #BLOCKA,%4      ;SET LINE ADDRESS IN R4.
005632  104002                  TYPLB:  STALL                   ;STALL IF ALLOWED.
005634  112400                          MOVB    (4)+,%0         ;GET CHARACTER
005636  004767  177622                  JSR     %7,LSPCH        ;GO OUTPUT CHARACTER.
005642  005367  000030                  DEC     TCTR            ;DONE?
005646  001371                          BNE     TYPLB           ;BRANCH IF NOT DONE.
005650  112767  000000  176624  TYPCLR: MOVB    #0,TYPDAT       ;OUTPUT 2 NULLS
005656  004767  176510                  JSR     %7,OUTTYP       ;AT END OF MESSAGE
005662  112767  000000  176612          MOVB    #0,TYPDAT       ;TO CLEAR A
005670  004767  176476                  JSR     %7,OUTTYP       ;DOUBLE BUFFER
005674  000207                          RTS     %7              ;DONE, EXIT
005676  000000          TCTR:   OPEN
                                ;SUBROUTINE TO TYPE LINE OF 3 CHARACTERS
005700  011667  000016          TYPL3:  MOV     @%6,TPL3A       ;DEVELOP AND SET ADDRESS OF
005704  017767  000012  000010          MOV     @TPL3A,TPL3A    ;DATA IN TPL3A.
005712  062716  000002                  ADD     #2,@%6          ;SET UP EXIT.
005716  004567  000034                  JSR     %5,FBF3         ;FILL BUFFER WITH 3 CHARACTERS
005722  000000          TPL3A:  OPEN
005724  042767  040000  173420          BIC     #BIT14,@PRGID   ;DISABLE STALLS.
005732  004767  177662                  JSR     %7,TYPLN        ;GO TYPE LINE OF CHARACTERS.
005736  000002                          RTI                     ;EXIT.
005740  112767  000015  006607  STBF:   MOVB    #15,BLOCKA      ;SUB TO SET UP BUFFER AREA.
005746  112767  000012  006602          MOVB    #12,BLOCKA+1
005754  000207                          RTS     %7              ;EXIT
                                ;SUBROUTINE TO FILL CHARACTER BUFFER WITH 3 CHARACTERS.
005756  012567  000004          FBF3:   MOV     (5)+,FBF3A
005762  004567  177440                  JSR     %5,BMOVE        ;MOVE 3 CHARS TO BUFFER.
005766  000000          FBF3A:  OPEN
005770  014557                          BLOCK1
005772  000003                          3
005774  004567  177426          FBF3B:  JSR     %5,BMOVE        ;FILL 72 CHARACTERS BUFFER
006000  014557                          BLOCK1                  ;WITH 3 CHARACTERS
006002  014562                          BLOCK1+3
006004  000110                          72.
006006  004567  177414                  JSR     %5,BMOVE
006012  014557                          BLOCK1
006014  014671                          BLOCK2
006016  000120                          80.
006020  000205                          RTS     %5              ;EXIT

                                ;SUBROUTINE TO FILL BUFFER WITH ALL CHARACTERS
006022  004567  177400          FBALL:  JSR     %5,BMOVE        ;FILL 72 CHAR BUFFER WITH
006026  013402                          A                       ;ALL CHARACTERS
006030  014557                          BLOCK1
006032  000063                          63
006034  004567  177366                  JSR     %5,BMOVE
006040  013402                          A
006042  014656                          BLOCK1+63.
006044  000022                          18.
006046  004567  177354                  JSR     %5,BMOVE
006052  014557                          BLOCK1
006054  014671                          BLOCK2
```

```
PAL,X11  V003    28-JUN-72        17:07    PAGE 39-1

004056  000120                      99:
004060  000207                      RTS     X7              ;EXIT;
```

```
                                ;PRG0 - INPUT-OUTPUT LOGIC TESTS
006062  012767  006074  173246  PRG01:  MOV     #AT0,KSTART     ;ADDRESS OF 1ST ROUTINE TO KSTART.
006070  000167  174316          JMP     SRSET           ;GO GET STARTED.
                                ;TEST ABILITY TO REFERENCE THE KEYBOARD/READER STATUS WORD (TKS)
006074  000000          AT0:    0                       ;TEST #.
006076  006126                  AT1                     ;NEXT TEST.
006100  001750                  1000.                   ;1 COUNT.
006102  006112                  AT0A                    ;SCOPE ENTRY.
006104  012767  006122  171672          MOV     #AT0E,MACHER    ;SET UP MACHINE ERROR TRAP.
006112  005777  173174  AT0A:   TST     @TKS            ;REFERENCE CODER STATUS WORD.
006116  104012          AT0B:   CHAIN                   ;CHAIN
006120  000774                  BR      AT0A            ;REPEAT TEST.
006122  104003          AT0E:   ERROR                   ;'ERROR' TRAPPED WHEN REFERENCING READER
006124  000774                  BR      AT0B            ;STATUS WORD (TKS).
                                ;TEST ABILITY TO REFERENCE THE KEYBOARD/READER BUFFER (TKB).
006126  000001          AT1:    1                       ;TEST #.
006130  006160                  AT2                     ;NEXT TEST.
006132  001750                  1000.                   ;1 COUNT.
006134  006144                  AT1A                    ;SCOPE ENTRY.
006136  012767  006154  171640          MOV     #AT1E,MACHER    ;SET UP MACHINE ERROR TRAP
006144  005777  173144  AT1A:   TST     @TKB            ;REFERENCE READER BUFFER.
006150  104012          AT1B:   CHAIN                   ;CHAIN
006152  000774                  BR      AT1A            ;REPEAT TEST.
006154  104003          AT1E:   ERROR                   ;'ERROR' TRAPPED WHEN REFERENCING
006156  000774                  BR      AT1B            ;READER BUFFER (TKB).
                                ;TEST ABILITY TO REFERENCE PUNCH/PRINTER STATUS WORD (TPS).
006160  000002          AT2:    2                       ;TEST #.
006162  006212                  AT3                     ;NEXT TEST
006164  001750                  1000.                   ;1 COUNT.
006166  006176                  AT2A                    ;SCOPE ENTRY.
006170  012767  006206  171606          MOV     #AT2E,MACHER    ;SETUP MACHINE ERROR TRAP.
006176  005777  173114  AT2A:   TST     @TPS            ;REFERENCE PUNCH/PRINTER STATUS WORD.
006202  104012          AT2B:   CHAIN                   ;CHAIN
006204  000774                  BR      AT2A            ;REPEAT TEST.
006206  104003          AT2E:   ERROR                   ;'ERROR' TRAPPED WHEN REFERENCING
006210  000774                  BR      AT2B            ;PUNCH/PRINTER STATUS WORD (TPS).
                                ;TEST ABILITY TO REFERENCE PUNCH/PRINTER BUFFER (TPB).
006212  000003          AT3:    3
006214  006244                  AT4
006216  001750                  1000.
006220  006230                  AT3A
006222  012767  006240  171554          MOV     #AT3E,MACHER    ;SETUP MACHINE ERROR TRAP.
006230  005777  173064  AT3A:   TST     @TPB            ;REFERENCE PUNCH/PRINTER BUFFER.
006234  104012          AT3B:   CHAIN                   ;CHAIN
006236  000774                  BR      AT3A            ;REPEAT TEST.
006240  104003          AT3E:   ERROR                   ;'ERROR' TRAPPED WHEN REFERENCING
```

```
006242  000774                          BR      AT3B            ;PUNCH?PRINTER BUFFER, (TPS);
                                ;TEST ABILITY TO SET AND CLEAR READER/KYBD ID BIT
006244  000004                 AT4:    4                       ;TEST #
006246  006330                         AT5                     ;NEXT TEST
006250  001750                         1000.                   ;I COUNT
006252  006262                         AT4A                    ;SCOPE ENTRY
006254  012767  000340  171514        MOV     #PRTY7,PSW      ;SET PRIORITY 7,
006262  052777  000100  173022 AT4A1:  BIS     #BIT6,@TKS      ;SET ID BIT IN TKS;
006270  032777  000100  173014        BIT     #BIT6,@TKS      ;CHECK ID BIT IN TKS
006276  001002                         BNE     AT4B            ;BRANCH IF ID BIT IS SET,
006300  104003                 AT4E1:  ERROR                   ;ERROR 1 ID BIT NOT SET;
006302  000410                         BR      AT4C
006304  042777  000100  173000 AT4B1:  BIC     #BIT6,@TKS      ;CLEAR ID BIT IN TKS
006312  032777  000100  172772        BIT     #BIT6,@TKS      ;CHECK ID BIT IN TKS;
006320  001401                         BEQ     AT4C            ;BRANCH IF ID BIT IS CLEARED;
006322  104003                 AT4E2:  ERROR                   ;ERROR; ID BIT FAILED TO CLEAR;
006324  104012                 AT4C1:  CHAIN                   ;CHAIN
006326  000755                         BR      AT4A            ;REPEAT TEST,
                                ;TEST ABILITY TO CLEAR ID BIT WITH RESET INSTRUCTION,
006330  000005                 AT5:    5                       ;TEST #
006332  007220                         AT24                    ;NEXT TEST
006334  000144                         100.                    ;I COUNT
006336  006346                         AT5A                    ;SCOPE ENTRY,
006340  012767  000340  171430        MOV     #PRTY7,PSW      ;SET PRIORITY 7,
006346  052777  000100  172736 AT5A1:  BIS     #BIT6,@TKS      ;SET ID BIT IN TKS
006354  104011                         SRESET                  ;RESET
006356  032777  000100  172726        BIT     #BIT6,@TKS      ;TEST ID BIT,
006364  001401                         BEQ     AT5B            ;BRANCH IF ID BIT IS CLEAR;
006366  104003                 AT5E1:  ERROR                   ;ERROR; RESET FAILED TO CLEAR ID BIT,
006370  104012                 AT5B1:  CHAIN                   ;CHAIN
006372  000765                         BR      AT5A            ;REPEAT TEST;
                                ;TEST THAT READER DONE BIT SETS BY 200 MSECS AFTER RDR ENABLE
006374  000011                 AT11:   11                      ;TEST #
006376  006406                         AT12                    ;NEXT TEST
006400  000144                         100.                    ;COUNT
006402  006414                         AT11A                   ;SCOPE ENTRY
006404  104400                 AT11A1: DELAY
006406  000236                         150.                    ;WAIT 150 MSECS
006410  004767  175334                 JSR     %7,AREAD        ;ENABLE READER
006414  104012                 AT11B1: CHAIN                   ;CHAIN
006416  000772                         BR      AT11A           ;REPEAT TEST
```

```
                        ;TEST THAT DONE BIT READS RELIABLY.
006420  000012          AT12:   12                      ;TEST #
006422  006472                  AT13                    ;NEXT TEST
006424  000012                  10.                     ;I COUNT
006426  006430                  AT12A                   ;SCOPE ENTRY.
006430  104400          AT12A:  DELAY                   ;WAIT 150 MSECS.
006432  000226                  150.
006434  004767  175310  AT12B:  JSR     X7,AREAD        ;ENABLE READER, COME BACK WHEN DONE SET.
006440  012767  001750  173020  MOV     #1000.,,CTRA    ;1000 TO CTRA
006446  105777  172640  AT12C:  TSTB    @TKS            ;TEST FOR DONE
006452  100402                  BMI     AT12D           ;BRANCH IF DONE FOUND SET.
006454  104003          AT12E:  ERROR                   ;ERROR, DONE BIT NOT FOUND SET.
006456  000403                  BR      AT12F
006460  005367  173002  AT12D:  DEC     CTRA            ;CHECKED 1000 TIMES?
006464  001370                  BNE     AT12C           ;BRANCH IF NOT 1000 YET.
006466  104012          AT12F:  CHAIN                   ;CHAIN
006470  000761                  BR      AT12B           ;REPEAT TEST.
                        ;TEST THAT RESET CLEARS DONE BIT
006472  000013          AT13:   13                      ;TEST #
006474  006530                  AT14                    ;NEXT TEST
006476  000144                  100.                    ;I COUNT
006500  006502                  AT13A                   ;SCOPE ENTRY
006502  104400          AT13A:  DELAY                   ;WAIT 150 MSECS.
006504  000226                  150.
006506  004767  175236  AT13B:  JSR     X7,AREAD        ;ENABLE READER, COME BACK WHEN DONE SET.
006512  104011                  SRESET                  ;ISSUE RESET.
006514  105777  172572          TSTB    @TKS            ;TEST FOR DONE BIT
006520  100001                  BPL     AT13C           ;BRANCH IF DONE BIT RESET.
006522  104003          AT13E:  ERROR                   ;ERROR, RESET FAILED TO CLEAR DONE.
006524  104012          AT13C:  CHAIN                   ;CHAIN
006526  000767                  BR      AT13B           ;REPEAT TEST.
                        ;TEST THAT REFERENCING READER DATA BUFFER CLEARS DONE
006530  000014          AT14:   14                      ;TEST #
006532  006570                  AT17                    ;NEXT TEST
006534  000144                  100.                    ;I COUNT
006536  006540                  AT14A                   ;SCOPE ENTRY
006540  104400          AT14A:  DELAY                   ;WAIT 150 MSECS.
006542  000226                  150.
006544  004767  175200  AT14B:  JSR     X7,AREAD        ;ENABLE READER, RETURN WHEN DONE SET.
006550  105777  172540          TSTB    @TKB            ;REFERENCE READ BUFFER.
006554  105777  172532          TSTB    @TKS            ;TEST FOR DONE BIT
006560  100001                  BPL     AT14C           ;BRANCH IF DONE NOT SET.
006562  104003          AT14E:  ERROR                   ;ERROR, REFERENCE TO BUFFER DID NOT RESET DONE.
006564  104012          AT14C:  CHAIN                   ;CHAIN
006566  000766                  BR      AT14B           ;REPEAT TEST
                        ;TEST THAT READ BUFFER CAN BE READ RELIABLY.
006570  000017          AT17:   17                      ;TEST#
006572  006664                  AT20                    ;NEXT TEST
006574  000144                  100.                    ;I COUNT
006576  006600                  AT17A                   ;SCOPE ENTRY
006600  104400          AT17A:  DELAY                   ;WAIT 150 MSECS
006602  000226                  150.
006604  004767  175140  AT17B:  JSR     X7,AREAD        ;ENABLE READER, RETURN WHEN DONE SET.
006610  117767  172500  172637  MOVB    @TKB,CRBUF+1    ;BUFFER CONTENTS TO CRBUF+1
006616  012767  000144  172642  MOV     #100.,,CTRA     ;100 TO CTRA
```

```
006624  117767  172464  172622  AT17C:  MOVB    @TKB,CRBUF      ;BUFFER CONTENTS TO CRBUF
006632  126767  172616  172615          CMPB    CRBUF,CRBUF+1   ;COMPARE CONTENTS OF CRBUF AND CRBUF+1
006640  001404                          BEQ     AT17D           ;BRANCH IF SAME,
006642  016700  172606                  MOV     CRBUF,R0        ;NOT SAME, ERROR, HALT WITH 1ST READ CHAR
006646  000000          AT17E:  HALT                            ;IN DATA BYTES LEFT, SUBSEQUENT READ IN DATA BYTES RIGHT,
006650  000403                          BR      AT17F
006652  005367  172610  AT17D:  DEC     CTRA            ;WERE IF SAME, CHECKED 100 TIMES?
006656  001362                          BNE     AT17C           ;BRANCH IF NOT,
006660  104012          AT17F:  CHAIN                           ;CHAIN
006662  000750                          BR      AT17B           ;REPEAT TEST,
```

```
                                ;TEST THAT READER DONE BIT IS ABLE TO CAUSE INTERRUPT; IF THE INTERRUPT IS
                                ;SERVICED, IT WILL HAVE OCURRED AT CORRECT VECTOR;
006664  000020                AT20:   20                      ;TEST#
006666  006742                        AT21                    ;NEXT TEST
006670  001750                        1000.                   ;1 COUNT
006672  006710                        AT20B                   ;SCOPE ENTRY
006674  104006                        STRDRV                  ;SET UP READER VECTOR TO AT20C
006676  006734                        AT20C
006700  104400                AT20A:  DELAY                   ;WAIT 150 MSECS.
006702  000226                        150.
006704  004767  175040                JSR     %7,AREAD        ;ENABLE READER; RETURN WHEN DONE SET.
006710  005077  172376        AT20B:  CLR     @TKS            ;DISABLE READER INTERRUPTS
006714  005067  171056                CLR     PSW             ;ENABLE READER; RETURN WHEN DONE SET.
006720  052777  000100  172364        BIS     #BIT6,@TKS      ;ENABLE READER INTERRUPT.
006726  000240                        NOP
006730  104003                AT20E:  ERROR                   ;ERROR; READER FAILED TO INTERRUPT;
006732  000401                        BR      AT20D
006734  022626                AT20C:  POPSP2                  ;HERE IF INTERRUPT OCCURS; POP STACK TWICE;
006736  104012                AT20D:  CHAIN                   ;CHAIN
006740  000763                        BR      AT20B           ;REPEAT TEST.
                                ;TEST THAT DONE DOES NOT CAUSE INTERRUPT WITH PROCESSOR AT SAME
                                ;PRIORITY LEVEL AS THE READERS INTERRUPT REQUEST LEVEL;
006742  000021                AT21:   21                      ;TEST#
006744  007026                        AT22                    ;NEXT TEST;
006746  001750                        1000.                   ;1 COUNT.
006750  006766                        AT21B                   ;SCOPE ENTRY.
006752  104006                        STRDRV                  ;SET READER VECTOR TO AT21E;
006754  007020                        AT21E
006756  104400                AT21A:  DELAY                   ;WAIT 150 MSECS.
006760  000226                        150.
006762  004767  174762                JSR     %7,AREAD        ;ENABLE READER; RETURN WHEN DONE SET.
006766  005077  172320        AT21B:  CLR     @TKS            ;DISABLE READER INTERRUPTS;
006772  016767  172326  170776        MOV     TKLVL,PSW       ;SET PROCESSOR TO SAME PRIORITY AS READER'S.
007000  052777  000100  172304        BIS     #BIT6,@TKS      ;ENABLE READER INTERRUPTS;
007006  000240                        NOP                     ;NO OP;
007010  005077  172276        AT21C:  CLR     @TKS            ;OK IF NO INTERRUPT OCCURS;
007014  104012                        CHAIN                   ;CHAIN
007016  000763                        BR      AT21B           ;REPEAT TEST.
007020  022626                AT21E:  POPSP2                  ;ERROR; READER ERRONEOUSLY INTERRUPTED
                                                              ;WITH PROCESSOR AT SAME PRIORITY
007022  104003                        ERROR                   ;LEVEL AS THE READER, OR THE READER
007024  000771                        BR      AT21C           ;IS AT HIGHER PRIORITY THAN SPECIFIED
                                                              ;AT TKLVL;


                                ;TEST THAT DONE CAUSES INTERRUPT WITH PROCESSOR AT PRIORITY ONE LEVEL LOWER
                                ;THAN THE READER'S INTERRUPT PRIORITY LEVEL.
007026  000022                AT22:   22                      ;TEST#
007030  007120                        AT23                    ;NEXT TEST
007032  001750                        1000.                   ;1 COUNT
007034  007106                        AT22B                   ;SCOPE ENTRY
007036  104006                        STRDRV                  ;SET READER INTERRUPT SERVICE TO
007040  007106                        AT22B                   ;AT22B;
007042  104400                        DELAY                   ;WAIT 150 MSECS.
```

```
007044  000226                          150,
007046  004767  174676                  JSR     %7,AREAD        ;ENABLE READER, RETURN WHEN DONE SET,
007052  005077  172234          AT22A:  CLR     @TKS            ;DISABLE READER INTERRUPTS
007056  016767  172242  170712          MOV     TKLVL,PSW       ;SET PROCESSOR PRIORITY ONE LEVEL LOWER
007064  162767  000040  170704          SUB     #40,PSW         ;TRAN READER,(SPECIFIED AT TKLVL),
007072  052777  000100  172212          BIS     #BIT6,@TKS      ;ENABLE READER INTERRUPTS,
007100  000240                          NOP
007102  104003                  AT22E:  ERROR                   ;FAILED TO INTERRUPT WITH PC AT PRIORITY ONE LEVEL LOWER
007104  000401                          BR      AT22C           ;TRAN READER, THEREFORE, READER PRIORITY MUST BE LOWER
007106  022626                  AT22B:  POPSP2                  ;HERE IF INTERRUPT OCCURS, OK, POP STACK TWICE
007110  005077  172176          AT22C:  CLR     @TKS            ;DISABLE READER INTERRUPTS
007114  104012                          CHAIN                   ;CHAIN
007116  000755                          BR      AT22A           ;REPEAT TEST
                                ;TEST THAT DONE DOES NOT REINTERRUPT AFTER RTI WHEN DONE IS NOT CLEARED,
007120  000023                  AT23:   23                      ;TEST#
007122  010294                          AT43                    ;NEXT TEST
007124  001750                          1000,                   ;1 COUNT
007126  007140                          AT23A                   ;SCOPE ENTRY,
007130  104400                          DELAY                   ;WAIT 150 MSECS
007132  000226                          150,
007134  004767  174610                  JSR     %7,AREAD        ;ENABLE READER, RETURN WHEN DONE SET,
007140  104006          AT23A:  STRDRV                          ;SET READER INTERRUPT SERVICE
007142  007172                          AT23C                   ;TO AT23C,
007144  005077  172142                  CLR     @TKS            ;DISABLE READER INTERRUPTS,
007150  052777  000100  172134          BIS     #BIT6,@TKS      ;ENABLE READER INTERRUPTS,
007156  000240                          NOP
007160  104003          AT23E1: ERROR                           ;ERROR1 FAILED TO INTERRUPT
007162  005077  172124          AT23B:  CLR     @TKS            ;DISABLE READER INTERRUPTS,
007166  104012                          CHAIN                   ;CHAIN
007170  000763                          BR      AT23A           ;REPEAT TEST
007172  012777  007212  172122  AT23C:  MOV     #AT23E2,@TKVTR  ;CHANGE INTERRUPT VECTOR TO AT23E2
007200  012716  007206                  MOV     #AT23D,@X6
007204  000002                          RTI                     ;RETURN FROM INTERRUPT
007206  000240          AT23D:  NOP
007210  000764                          BR      AT23B           ;OK IF NO REINTERRUPT OCCURS,
007212  022626          AT23E2: POPSP2                          ;ERROR2, DONE REINTERRUPTED AFTER
007214  104003                  ERROR                           ;RTI WITH DONE BIT LEFT ON,
007216  000761                  BR      AT23B
```

```
                                        ;TEST ABILITY TO SET AND CLEAR PUNCH ID BIT
007220  000024                AT24:   24                      ;TEST#
007222  007304                        AT25                    ;NEXT TEST,
007224  001750                        1000.                   ;I COUNT
007226  007236                        AT24A                   ;SCOPE ENTRY,
007230  012767  000340  170540        MOV     #PRTY7,PSW      ;SET PRIORITY 7,
007236  052777  000100  172052 AT24A: BIS     #BIT6,@TPS      ;SET PUNCH ID BIT,
007244  032777  000100  172044        BIT     #BIT6,@TPS      ;CHECK PUNCH ID BIT,
007252  001002                        BNE     AT24B           ;BRANCH IF PUNCH ID BIT IS SET,
007254  104003                AT24E1: ERROR                   ;ERROR1, PUNCH ID BIT DID NOT SET,
007256  000410                        BR      AT24C
007260  042777  000100  172030 AT24B: BIC     #BIT6,@TPS      ;CLEAR PUNCH ID BIT,
007266  032777  000100  172022        BIT     #BIT6,@TPS      ;CHECK PUNCH ID BIT,
007274  001401                        BEQ     AT24C           ;BRANCH IF PUNCH ID BIT IS CLEAR
007276  104003                AT24E2: ERROR                   ;ERROR2, PUNCH ID BIT FAILED TO CLEAR,
007300  104012                AT24C:  CHAIN                   ;CHAIN
007302  000776                        BR      AT24C           ;REPEAT TEST
                                        ;TEST ABILITY TO CLEAR PUNCH ID BIT WITH RESET INSTRUCTION
007304  000025                AT25:   25                      ;TEST#
007306  007350                        AT30                    ;NEXT TEST,
007310  000144                        100.                    ;I COUNT,
007312  007322                        AT25A                   ;SCOPE ENTRY,
007314  012767  000340  170454        MOV     #PRTY7,PSW      ;SET PRIORITY 7,
007322  052777  000100  171766 AT25A: BIS     #BIT6,@TPS      ;SET PUNCH ID BIT,
007330  104011                        SRESET                  ;RESET
007332  032777  000100  171756        BIT     #BIT6,@TPS      ;CHECK PUNCH ID BIT,
007340  001401                        BEQ     AT25B           ;BRANCH IF PUNCH ID BIT IS CLEAR,
007342  104003                AT25E1: ERROR                   ;ERROR, RESET FAILED TO CLEAR PUNCH ID BIT,
007344  104012                AT25B:  CHAIN                   ;CHAIN
007346  000765                        BR      AT25A           ;REPEAT TEST,
```

```
007350  000030                  ;TEST THAT RESET SETS THE PUNCH READY BIT, AND THAT READY CAN BE READ RELIABLY;
007352  007374          AT30:   30                      ;TEST#
007354  001750                  AT31                    ;NEXT TEST
007356  007360                  1000.                   ;# COUNT
007360  105777  171732          AT30A                   ;SCOPE ENTRY
007364  100401          AT30A:  TSTB    @TPS            ;CHECK PUNCH READY;
007366  104003                  BMI     AT30B           ;BRANCH IF PUNCH READY IS SET;
007370  104012          AT30E:  ERROR                   ;ERROR; RESET FAILED TO SET READY, OR FAILED TO READ  IT;
007372  000772          AT30B:  CHAIN                   ;CHAIN
                                BR      AT30A           ;REPEAT TEST.
                        ;TEST THAT PUNCH READY RESETS BY LOADING PUNCH BUFFER;
007374  000031          AT31:   31                      ;TEST#
007376  007432                  AT32                    ;NEXT TEST
007400  000024                  20.                     ;# COUNT
007402  007404                  AT31A                   ;SCOPE ENTRY
007404  104400          AT31A:  DELAY                   ;WAIT 150 MSECS
007406  000226                  150.
007410  104011                  SRESET                  ;RESET
007412  005077  171702          CLR     @TPB            ;LOAD PUNCH BUFFER
007416  105777  171674          TSTB    @TPS            ;CHECK PUNCH READY BIT;
007422  100001                  BPL     AT31B           ;BRANCH IF PUNCH READY IS CLEAR.
007424  104003          AT31E:  ERROR                   ;ERROR; BUFFER LOAD FAILED TO CLEAR READY;
007426  104012          AT31B:  CHAIN                   ;CHAIN
007430  000765                  BR      AT31A           ;REPEAT TEST.
                        ;TEST THAT BYTE LOAD OF PUNCH BUFFER +1 DOES NOT RESET READY;
007432  000032          AT32:   32                      ;TEST#
007434  007474                  AT33                    ;NEXT TEST
007436  000024                  20.                     ;# COUNT
007440  007442                  AT32A                   ;SCOPE ENTRY
007442  104400          AT32A:  DELAY                   ;WAIT 150 MSECS
007444  000226                  150.
007446  104011                  SRESET                  ;RESET
007450  016700  171644          MOV     TPB,R0
007454  005200                  INC     R0
007456  105010                  CLRB    @R0             ;BYTE LOAD PUNCH BUFFER+1
007460  105777  171632          TSTB    @TPS            ;CHECK PUNCH READY BIT
007464  100401                  BMI     AT32B           ;BRANCH IF PUNCH READY STILL SET;
007466  104003          AT32E:  ERROR                   ;ERROR; BYTE LOAD OF PUNCH BUFFER+1
007470  104012          AT32B:  CHAIN                   ;CLEARED READY; CHAIN
007472  000763                  BR      AT32A           ;REPEAT TEST.
```

```
                                        ;TEST THAT PUNCH BECOMES READY BY 200 MSECS AFTER BUFFER LOAD;
007474  000033                  AT331:  33                      ;TEST #
007476  007534                          AT34                    ;NEXT TEST
007500  000024                          20;                     ;I COUNT
007502  007504                          AT33A                   ;SCOPE ENTRY;
007504  104400                  AT33A:  DELAY                   ;WAIT 150 MSECS;
007506  000226                          150;
007510  005077  171604                  CLR     @TPB            ;LOAD PUNCH BUFFER;
007514  104400                          DELAY                   ;WAIT 200 MSECS;
007516  000310                          200;
007520  105777  171572                  TSTB    @TPS            ;CHECK PUNCH READY BIT;
007524  100401                          BMI     AT33B           ;BRANCH IF PUNCH READY IS SET;
007526  104003                  AT33E:  ERROR                   ;ERROR; READY NOT SET 200 MSECS AFTER BUFFER LOAD;
007530  104012                  AT33B:  CHAIN                   ;CHAIN
007532  000764                          BR      AT33A           ;REPEAT TEST;
                                        ;TEST THAT PUNCH READY BIT CAN CAUSE AN INTERRUPT; IF THE INTERRUPT
                                        ;IS SERVICED, IT WILL HAVE OCCURRED AT THE CORRECT VECTOR;
007534  000034                  AT341:  34                      ;TEST #
007536  007602                          AT35                    ;NEXT TEST
007540  001750                          1000;                   ;I COUNT
007542  007550                          AT34A                   ;SCOPE ENTRY
007544  104007                          STPCHV                  ;SET PUNCH INTERRUPT SERVICE
007546  007576                          AT34C                   ;TO AT34C
007550  005077  171542          AT34A:  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
007554  005067  170216                  CLR     PSW             ;SET PRIORITY 0;
007560  052777  000100  171530          BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS;
007566  000240                          NOP
007570  104003                  AT34E:  ERROR                   ;PUNCH READY FAILED TO CAUSE
007572  104012                  AT34B:  CHAIN                   ;INTERRUPT; CHAIN
007574  000765                          BR      AT34A           ;REPEAT TEST;
007576  022626                  AT34C:  POPSP2                  ;HERE IF INTERRUPT OCCURS; POP THE
007600  000774                          BR      AT34B           ;STACK TWICE;
                                        ;TEST THAT PUNCH READY DOES NOT CAUSE AN INTERRUPT WITH PROCESSOR
                                        ;AT SAME PRIORITY LEVEL AS THE PUNCH INTERRUPT REQUEST LEVEL;
007602  000035                  AT351:  35                      ;TEST #
007604  007656                          AT36                    ;NEXT TEST
007606  001750                          1000;                   ;I COUNT
007610  007616                          AT35A                   ;SCOPE ENTRY
007612  104007                          STPCHV                  ;SET PUNCH INTERRUPT SERVICE
007614  007650                          AT35E                   ;TO AT35E;
007616  016767  171506  170152  AT35A:  MOV     TPLVL,PSW       ;SET PROCESSOR TO SAME PRIORITY AS PUNCH;
007624  005077  171466                  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS;
007630  052777  000100  171460          BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS;
007636  000240                          NOP
007640  005077  171452          AT35B:  CLR     @TPS            ;OK IF NO INTERRUPT OCCURS;
007644  104012                          CHAIN                   ;CHAIN
007646  000763                          BR      AT35A           ;REPEAT TEST;
007650  022626                  AT35E:  POPSP2                  ;ERROR; PUNCH INTERRUPTED WITH PROCESSOR
007652  104003                          ERROR                   ;SET TO SAME PRIORITY AS THE PUNCH;
```

```
007654  000771                  BR      AT35B
                        ;TEST THAT THE PUNCH INTERRUPTS WITH PROCESSOR AT PRIORITY ONE LEVEL LOWER
                        ;THAN THE PUNCH PRIORITY.
007656  000036          AT36:   36                      ;TEST #
007660  007740                  AT37                    ;NEXT TEST
007662  001750                  1000.                   ;I COUNT
007664  007672                  AT36A                   ;SCOPE ENTRY
007666  104007                  STPCHV                  ;SET PUNCH INTERRUPT SERVICE
007670  007726                  AT36B                   ;TO AT36B.
007672  005077  171420  AT36A:  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
007676  016767  171426  170072  MOV     TPLVL,PSW       ;SET PROCESSOR PRIORITY ONE LEVEL
007704  162767  000040  170064  SUB     #40,PSW         ;LOWER THAN PUNCH PRIORITY
007712  052777  000100  171376  BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
007720  000240                  NOP
007722  104003                  ERROR                   ;ERROR, PUNCH FAILED TO INTERRUPT.
007724  000401                  BR      AT36C
007726  022626          AT36B:  POPSP2                  ;HERE IF INTERRUPT OCCURS, POP
007730  005077  171362  AT36C:  CLR     @TPS            ;THE STACK TWICE, DISABLE PUNCH INTERRUPT
007734  104012                  CHAIN                   ;CHAIN
007736  000755                  BR      AT36A           ;REPEAT TEST.
                        ;TEST THAT PUNCH READY DOES NOT REINTERRUPT AFTER RTI WHEN READY
                        ;BIT HAS NOT BEEN RESET.
007740  000037          AT37:   37                      ;TEST #
007742  010034                  AT48                    ;NEXT TEST
007744  001750                  1000.                   ;I COUNT
007746  007750                  AT37A                   ;SCOPE ENTRY
007750  104007          AT37A:  STPCHV                  ;SET PUNCH INTERRUPT SERVICE TO
007752  010006                  AT37C                   ;AT37C
007754  005077  171336          CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
007760  005067  170012          CLR     PSW             ;SET PROCESSOR PRIORITY TO 0
007764  052777  000100  171324  BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
007772  000240                  NOP
007774  104003          AT37E1: ERROR                   ;ERROR 1, PUNCH FAILED TO INTERRUPT.
007776  005077  171314  AT37B:  CLR     @TPS            ;DISABLE PUNCH INTERRUPT.
010002  104012                  CHAIN                   ;CHAIN
010004  000761                  BR      AT37A           ;REPEAT TEST.
010006  012777  010022  171312  AT37C:  MOV     #AT37E2,@TPVTR  ;HERE IF INTERRUPT OCCURS, CHANGE
010014  012716  010022          MOV     #AT37D,@X6      ;PUNCH VECTOR TO AT37E2 AND EXIT
010020  000002                  RTI                     ;INTERRUPT
010022  000240          AT37D:  NOP                     ;OK IF NO REINTERRUPT OCCURS
010024  000764                  BR      AT37B
010026  022626          AT37E2: POPSP2                  ;ERROR 2.  PUNCH REINTERRUPTED AFTER
010030  104003                  ERROR                   ;RTI WITH READY BIT LEFT ON
010032  000761                  BR      AT37B
```

```
                                ;TEST THAT THE PUNCH INTERRUPTS IMMEDIATELY UPON LOWERING
                                ;PROCESSOR PRIORITY TO 0;
010034  000040                  AT401:  40                      ;TEST #
010036  010120                          AT41                    ;NEXT TEST
010040  001750                          1000.                   ;I COUNT
010042  010050                          AT40A                   ;SCOPE ENTRY
010044  104007                          STPCHV                  ;SET PUNCH INTERRUPT
010046  010106                          AT40B                   ;SERVICE TO AT40B
010050  012767  000340  167720  AT40A:  MOV     #PRTY7,PSW      ;SET PROCESSOR PRIORITY TO 7;
010056  005077  171234                  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
010062  052777  000100  171226          BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
010070  005067  167702                  CLR     PSW             ;LOWER PROCESSOR PRIORITY TO 0;
010074  012767  000340  167674          MOV     #PRTY7,PSW      ;RAISE PRIORITY TO 7;
010102  104003                          ERROR                   ;ERROR; PUNCH FAILED TO INTERRUPT
010104  000401                          BR      AT40C           ;IMMEDIATELY AFTER CP PRIORITY WAS SET TO 0,
010106  022626                  AT40B:  POPSP2                  ;HERE IF INTERRUPT OCCURS
010110  005077  171202          AT40C:  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
010114  104012                          CHAIN                   ;CHAIN
010116  000754                          BR      AT40A           ;REPEAT TEST
                                ;TEST FOR CORRECT OPERATION OF THE WAIT INSTRUCTION.  A WAIT INSTRUCTION
                                ;IS PERFORMED WHILE WAITING FOR A PUNCH INTERRUPT;  WHEN THE INTERRUPT
                                ;OCCURS, THE SERVICE ROUTINE CHANGES THE WAIT INSTRUCTION TO AN ERROR
                                ;CALL AND THEN EXITS THE INTERRUPT WITH AN RTI;  EXITING THE INTERRUPT
                                ;SHOULD RETURN CONTROL TO THE INSTRUCTION FOLLOWING THE WAIT INSTRUCTION,
                                ;IF CONTROL IS INSTEAD RETURNED TO THE SAME LOCATION WHERE THE WAIT
                                ;INSTRUCTION WAS LOCATED AN ERROR CALL WILL OCCUR, INDICATING A FAILURE
                                ;OF THE WAIT INSTRUCTION;
010120  000041                  AT411:  41                      ;TEST#
010122  010206                          AT42                    ;NEXT TEST
010124  000062                          50;                     ;I COUNT
010126  010140                          AT41A                   ;SCOPE ENTRY
010130  104408                          DELAY                   ;WAIT 150 MSECS
010132  000226                          150.
010134  104007                          STPCHV                  ;SET PUNCH INTERRUPT SERVICE
010136  010176                          AT41C                   ;TO AT41C
010140  012767  000001  000016  AT41A:  MOV     #WAIT,AT41B     ;MOVE WAIT INSTRUCTION TO AT41B
010146  005077  171146                  CLR     @TPB            ;LOAD PUNCH BUFFER (ENABLES PUNCH)
010152  052777  000100  171136          BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
010160  005067  167712                  CLR     PSW             ;SET PRIORITY 0,
010164  000000          AT41B:  OPEN                            ;THIS LOCATION CAN BE EITHER
                                                                ;A WAIT INSTRUCTION OR AN ERROR CALL;
                                                                ;IF AN ERROR CALL IS EXECUTED, IT
                                                                ;INDICATES A FAILURE OF THE WAIT INSTRUCTION;
010166  005077  171124                  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
010172  104012                          CHAIN                   ;CHAIN
010174  000761                          BR      AT41A           ;REPEAT TEST
010176  012767  104003  177760  AT41C:  MOV     #ERROR,AT41B    ;MOVE ERROR CALL TO AT41B;
010204  000002                          RTI                     ;EXIT INTERRUPT;
```

```
                                        ;TEST THAT LOADING THE PUNCH BUFFER WITH THE MAINTENANCE BIT SET
                                        ;CAUSES THE READER DONE BIT TO SET AFTER APPROX, 200 MSECS
010206  000042                 AT421:  42                      ;TEST #
010210  004374                         AT11                    ;NEXT TEST
010212  000002                         2$,                     ;; COUNT
010214  010222                         AT42A                   ;SCOPE ENTRY
010216  104400                         DELAY
010220  000226                         100,
010222  052777  000004  171066 AT42A1: BIS     #BIT2,@TPS      ;SET MAINTENANCE BIT
010230  005077  171064                 CLR     @TPB            ;LOAD PUNCH BUFFER
010234  104400                         DELAY                   ;WAIT 200 MSECS
010236  000310                         200,
010240  105777  171046                 TSTB    @TKS            ;TEST READER DONE BIT
010244  100401                         BMI     AT42B           ;BRANCH IF READER DONE BIT SET;
010246  104003                 AT42E1: ERROR                   ;ERROR, 200 MSECS AFTER PUNCH
                                        ;BUFFER LOAD WITH MAINTENANCE BIT
                                        ;SET THE READER DONE BIT WAS NOT SET
010250  104012                 AT42B1: CHAIN                   ;CHAIN
010252  000763                         BR      AT42A           ;REPEAT TEST
                                        ;TESTS THAT DATA CAN BE RELIABLY
                                        ;TRANSFERRED WITH THE MAINTENANCE BIT
                                        ;SET;  CAUSES PROGRAM 7 TO BE
                                        ;EXECUTED AS A SUBROUTINE IF THIS
                                        ;WAS A MONITOR LOAD OR IF THE
                                        ;MONITOR CELL IS NON-ZERO
010254  000043                 AT431:  43                      ;TEST #
010256  177777                         -1                      ;LAST TEST
010260  001750                         1000,                   ;;COUNT
010262  012420                         LTA                     ;SCOPE ENTRY
010264  005767  167552                 TST     MONITR          ;IS THIS AUTOMATIC TESTING
010270  001402                         BEQ     AT43B           ;NO - EXIT
010272  004767  002102         AT43A1: JSR     X7,PRG7         ;RUN PROGRAM 7
010276  104012                 AT43B1: CHAIN                   ;CHAIN
010300  004767  002114                 JSR     X7,LTA          ;REPEAT TEST
010304  000774                         BR      AT43B
```

```
                                ;PRG1-PRINTER TESTS
010306  012767  010340  171022  PRG1:   MOV     #CT0,KSTART     ;SET ADDRESS IF 1ST ROUTINE.
010314  052767  000200  171030          BIS     #BIT7,PRGID     ;BYPASS SCOPE AND ;CNT';
010322  012767  177600  174310          MOV     #177600,STLMSK  ;SET STALL LIMIT
010330  004767  175404                  JSR     %7,STBF         ;SET UP BUFFER AREA,
010334  000167  172052                  JMP     SRSET           ;GO GET STARTED.
                                ;CARRIAGE RETURN TEST';
010340  000000                  CT0:    0                       ;TEST#
010342  010442                          CT1                     ;NEXT TEST ADDRESS';
010344  104000                          TYPE                    ;TYPE TITLE,
010346  013574                          CRTST
010350  012767  000120  171074          MOV     #80.,,RCNT
010356  016767  171070  171102          MOV     RCNT,CTRA       ;RCNT TO CTRA
010364  005367  171076          CT0A:   DEC     CTRA            ;DECREMENT CTRA
010370  001001                          BNE     CT0B            ;BRANCH IF NOT 0
010372  104012                          CHAIN                   ;0', CHAIN
010374  016767  171066  171066  CT0B:   MOV     CTRA,CTRB       ;SPACE COUNT TO CTRB';
010402  112700  000105          CT0C:   MOVB    #105,%0         ;CHARSE
010406  004767  175052                  JSR     %7,LSPCH        ;SPACE';
010412  005367  171052                  DEC     CTRB            ;DECREMENT CTRB,
010416  001371                          BNE     CT0C            ;BRANCH IF NOT DONE SPACING';
010420  112700  000015                  MOVB    #15,%0
010424  004767  175034                  JSR     %7,LSPCH        ;CARRIAGE RETURN,
010430  012700  000012                  MOV     #12,%0          ;LINE FEED
010434  004767  175024                  JSR     %7,LSPCH        ;
010440  000751                          BR      CT0A
                                ;RIGHT MARGIN TEST
010442  000001                  CT1:    1                       ;TEST#
010444  010506                          CT2                     ;NEXT TEST';
010446  104000                          TYPE                    ;TYPE TITLE
010450  013625                          RMTST
010452  012767  000024  171006          MOV     #20',,CTRA      ;SET UP FOR 33/35
010460  012767  013565  000014          MOV     #RM33B,RMB
010466  104000          CT1A:   TYPE                    ;TYPE--!
010470  013547                          RM33A
010472  005367  170770                  DEC     CTRA            ;DONE N TIMES';
010476  001373                          BNE     CT1A            ;BRANCH IF NOT N TIMES
010500  104000                          TYPE                    ;TYPE-!-,
010502  000000                  RMB:    0PEN
010504  104012                          CHAIN                   ;CHAIN';
```

```
                                        ;SPACE TEST
010576  000002                 CT2:   2                        ;TEST#
010510  010636                         CT3                     ;NEXT TEST
010512  104000                         TYPE                    ;TYPE TITLE,
010514  013653                         SPTST
010516  012767  000050  170742         MOV     #40.,CTRA       ;33/35 COUNT TO CTRA,
010524  104000                 CT2A:   TYPE                    ;TYPE SPACE,\,
010526  013671                         SPTSTC
010530  005367  170732                 DEC     CTRA            ;DONE TIMES SET IN CTRA?
010534  001373                         BNE     CT2A            ;BRANCH IF NOT DONE
010536  012767  000050  170722         MOV     #40.,CTRA       ;SET UP CTRA COUNT FOR 33/35
010544  012767  000001  170716 CT2B:   MOV     #1,CTRB
010552  010767  170712  170712 CT2C:   MOV     CTRB,CTRC
010560  112700  000015                 MOVB    #15,X0          ;CARRIAGE RETURN,
010564  004767  174074                 JSR     X7,LSPCH
010570  112700  000040         CT2D:   MOVB    #40,X0          ;CURSOR RIGHT
010574  004767  174064                 JSR     X7,LSPCH        ;SET IN CTRC,
010600  005367  170066                 DEC     CTRC            ;DONE SPACING,
010604  001371                         BNE     CT2D            ;BRANCH IF NOT DONE SPACING,
010606  112700  000057                 MOVB    #'/,X0          ;DONE, TYPE A "/",
010612  004767  174046                 JSR     X7,LSPCH
010616  005367  170044                 DEC     CTRA            ;DONE 36 TIMES?
010622  001001                         BNE     CT2E            ;BRANCH IF NOT DONE,
010624  104012                         CHAIN                   ;DONE, CHAIN,
010626  062767  000002  170634 CT2E:   ADD     #2,CTRB         ;MODIFY CTRB FOR NEXT TRY,
010634  000746                         BR      CT2C            ;GO DO IT AGAIN,
                                        ;LINE FEED TEST
010636  000003                 CT3:   3                        ;TEST #
010640  010710                         CT4                     ;NEXT TEST,
010642  104000                         TYPE                    ;TYPE TITLE
010644  013673                         LFTST
010646  052767  040000  170476         BIS     #BIT14,PRGID    ;ALLOW STALLS,
010654  012767  000120  170604         MOV     #80.,CTRA       ;SET 33/35 LINE FEED COUNT,
010662  112700  000134         CT3A:   MOVB    #'\,X0          ;TYPE "\"
010666  004767  174072                 JSR     X7,LSPCH
010672  112700  000012                 MOVB    #12,X0          ;LINE FEED,
010676  004767  174062                 JSR     X7,LSPCH
010702  005367  170060                 DEC     CTRA            ;DONE N TIMES?
010706  001001                         BNE     CT3B            ;BRANCH IF NOT DONE,
010710  104012                         CHAIN                   ;DONE, CHAIN
010712  104002                 CT3B:   STALL                   ;STALL
010714  000762                         BR      CT3A            ;REPEAT
```

```
                                        ;TAB TEST
010716  000004                  CT4:  4                       ;TEST#
010720  011130                        CT5                     ;NEXT TEST;
010722  012767  000011  000074        MOV     #9.,TBCNT       ;SET TAB COUNT;
010730  104014                        CK35                    ;35?
010732  104012                        CHAIN                   ;NO;
010734  004567  000040              JSR     %5,TPBM         ;TYPE MARKERS
010740  000007                      7
010742  104000                      TYPE
010744  013523                      TBMRK+1
010746  012767  000007  170512  CT4A: MOV     #7,CTRA         ;LINE COUNT TO CTRA
010754  005067  000046              CLR     SPCNT           ;0 TO SPACE COUNT;
010760  004767  000044        CT4B: JSR     %7,TABP         ;GO SPACE+TAB;
010764  005267  000036              INC     SPCNT           ;INCREMENT SPACE COUNT;
010770  005367  170472              DEC     CTRA            ;DONE 7 LINES?
010774  001371                      BNE     CT4B            ;BRANCH IF NOT DONE;
010776  104012                      CHAIN                   ;DONE; CHAIN;
011000  012567  170462        TPBM: MOV     (5)+,CTRA       ;TYPE TEST TITLE;
011004  104000                      TYPE
011006  013504                      TBTST
011010  104000                TPBMA: TYPE                   ;TYPE MARKERS
011012  013534                      TBMRK1
011014  005367  170446              DEC     CTRA
011020  001373                      BNE     TPBMA
011022  000205                      RTS     %5              ;EXIT;
011024  000000              TBCNT: OPEN                     ;TAB COUNT
011026  000000              SPCNT: OPEN                     ;SPACE COUNT
011030  104000                TABP: TYPE                    ;CRLF;
011032  013545                      CRLF
011034  016767  177764  170426        MOV     TBCNT,CTRB      ;TAB COUNT TO CTRB
011042  016767  177760  170422  TABPA: MOV     SPCNT,CTRC      ;SPACE COUNT TO CTRC
011050  001407                      BEQ     TABPC           ;BRANCH IF SPACE COUNT IS 0;
011052  112700  000040        TABPB: MOVB    #40,%0          ;SPACE
011056  004767  174402              JSR     %7,LSPCH
011062  005367  170404              DEC     CTRC            ;DECREMENT SPACE COUNT
011066  001371                      BNE     TABPB           ;BRANCH IF NOT YET 0;
011070  112700  000011        TABPC: MOVB    #11,%0          ;TAB
011074  004767  174364              JSR     %7,LSPCH
011100  004767  174360              JSR     %7,LSPCH        ;DUMMY CYCLE
011104  004767  174354              JSR     %7,LSPCH        ;DUMMY CYCLE;
011110  112700  000057              MOVB    #/,%0           ;TYPE "/"
011114  004767  174344              JSR     %7,LSPCH
011120  005367  170344              DEC     CTRB            ;DECREMENT TAB COUNT;
011124  001346                      BNE     TABPA           ;BRANCH IF NOT DONE TABBING;
011126  000207                      RTS     %7              ;DONE; EXIT;
```

```
011130  000005          ;TYPE LINE OF CHARACTERS ABC
011132  011140  CT5:     5                       ;TEST #
011134  104000          CT6                     ;NEXT TEST
011136  013710          TYPE                    ;TYPE "CHARACTER TESTS"
011140  104016          CHRTST
011142  013402          TYPLN3                  ;TYPE LINE
011144  104012          A
                        CHAIN                   ;CHAIN
                ;TYPE LINE OF CHARACTERS DEF
011146  000006  CT6:     6                       ;TEST #
011150  011160          CT7                     ;NEXT TEST
011152  104016          TYPLN3                  ;TYPE LINE
011154  013403          B
011156  104012          CHAIN                   ;CHAIN
                ;TYPE LINE OF CHARACTERS GHI
011160  000007  CT7:     7                       ;TEST#
011162  011172          CT10                    ;NEXT TEST;
011164  104016          TYPLN3                  ;TYPE LINE
011166  013404          C
011170  104012          CHAIN                   ;CHAIN
                ;TYPE LINE OF CHARACTERS OF JKL
011172  000010  CT10:    10                      ;TEST #
011174  011204          CT11                    ;NEXT TEST;
011176  104016          TYPLN3                  ;TYPELINE
011200  013405          D
011202  104012          CHAIN                   ;CHAIN
                ;TYPE LINE OF CHARACTERS MNO
011204  000011  CT11:    11                      ;TEST #
011206  011216          CT12                    ;NEXT TEST
011210  104016          TYPLN3                  ;TYPE LINE
011212  013406          E
011214  104012          CHAIN                   ;CHAIN
                ;TYPE LINE OF CHARACTERS PQR

011216  000012  CT12:    12                      ;TEST #
011220  011230          CT13                    ;NEXT TEST
011222  104016          TYPLN3                  ;TYPE LINE
011224  013407          F
011226  104012          CHAIN                   ;CHAIN
                ;TYPE LINE OF CHARACTERS STU
011230  000013  CT13:    13                      ;TEST #
011232  011242          CT14                    ;NEXT TEST
011234  104016          TYPLN3
011236  013410          G
011240  104012          CHAIN
```

```
                                ;TYPE LINE OF CHARACTERS VWX
011242  000014          CT141   14                      ;TEST #
011244  011254                  CT15                    ;NEXT TEST
011246  104016                  TYPLN3                  ;TYPE LINE
011250  013427                  V
011252  104012                  CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS Y20
011254  000015          CT151   15                      ;TEST #
011256  011266                  CT16                    ;NEXT TEST
011260  104016                  TYPLN3                  ;TYPE LINE
011262  013432                  V
011264  104012                  CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS 123
011266  000016          CT161   16                      ;TEST #
011270  011300                  CT17                    ;NEXT TEST
011272  104016                  TYPLN3                  ;TYPE LINE
011274  013435                  ONE
011276  104012                  CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS 456
011300  000017          CT171   17                      ;TEST #
011302  011312                  CT20                    ;NEXT TEST
011304  104016                  TYPLN3                  ;TYPE LINE
011306  013440                  FOUR
011310  104012                  CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS 789
011312  000020          CT201   20                      ;TEST #
011314  011324                  CT21                    ;NEXT TEST
011316  104016                  TYPLN3                  ;TYPE LINE
011320  013443                  SEVEN
011322  104012                  CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS!"#
011324  000021          CT211   21                      ;TEST #
011326  011336                  CT22                    ;NEXT TEST
011330  104016                  TYPLN3                  ;TYPE LINE
011332  013446                  C41
011334  104012                  CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS $%&
011336  000022          CT221   22                      ;TEST #
011340  011350                  CT23                    ;NEXT TEST
011342  104016                  TYPLN3                  ;TYPE LINE
011344  013451                  C44
011346  104012                  CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS '()
011350  000023          CT231   23                      ;TEST #
011352  011362                  CT24                    ;NEXT TEST
011354  104016                  TYPLN3                  ;TYPE LINE
011356  013454                  C47
011360  104012                  CHAIN                   ;CHAIN;
```

```
011362  000024
011364  011374
011366  104016
011370  013457
011372  104012

011374  000025
011376  011406
011400  104016
011402  013462
011404  104012

011406  000026
011410  011420
011412  104016
011414  013465
011416  104012

011420  000027
011422  011432
011424  104016
011426  013470
011430  104012

011432  000030
011434  011444
011436  104016
011440  013473
011442  104012

011444  000031
011446  011456
011450  104016
011452  013476
011454  104012
```

```
;TYPE LINE OF CHARACTERS *$,
CT241   24                      ;TEST #
        CT25                    ;NEXT TEST
        TYPLN3                  ;TYPE LINE
        C52
        CHAIN                   ;CHAIN
;TYPE LINE OF CHARACTERS *',/
CT251   25                      ;TEST #
        CT26                    ;NEXT TEST
        TYPLN3                  ;TYPE LINE
        C55
        CHAIN                   ;CHAIN
;TYPE LINE OF CHARACTERS !?<
CT261   26                      ;TEST #
        CT27                    ;NEXT TEST
        TYPLN3                  ;TYPE LINE
        C72
        CHAIN                   ;CHAIN
;TYPE LINE OF CHARACTERS &>?
CT271   27                      ;TEST #
        CT30                    ;NEXT TEST
        TYPLN3                  ;TYPE LINE
        C75
        CHAIN                   ;CHAIN',
;TYPE LINE OF CHARACTERS @[\
CT301   30                      ;TEST #
        CT31                    ;NEXT TEST
        TYPLN3                  ;TYPE LINE
        C100
        CHAIN                   ;CHAIN
;TYPE LINE OF CHARACTERS ]^AND LEFT ARROW
CT311   31                      ;TEST @
        CT32                    ;NEXT TEST
        TYPLN3                  ;TYPE LINE
        C135
        CHAIN                   ;CHAIN
```

```
                                        ;TYPE 20 LINES OF ALL CHARACTERS
                                        ;FIRST 10 FULL SPEED
                                        ;SECOND 10 AT STALL SPEED

011456  000032                  CT32:   32                      ;TEST #
011460  177777                          -1                      ;LAST TEST
011462  004767  174334                  JSR     X7,FBALL        ;FILL BUFFER WITH ALL CHARACTERS
011466  042767  040000  167656          BIC     #BIT14,PRGID    ;CLEAR STALL BIT
011474  012767  000012  000042          MOV     #10.,CNT                ;SET LINE COUNT = 10.
011502  004767  174112          CT32A:  JSR     X7,TYPLN        ;TYPE A LINE
011506  005367  000032                  DEC     CNT             ;
011512  001373                          BNE     CT32A           ;
011514  052767  040000  167630          BIS     #BIT14,PRGID    ;SET STALL
011522  012767  000012  000014          MOV     #10.,CNT                ;LINE COUNT = 10
011530  004767  174064          CT32B:  JSR     X7,TYPLN        ;TYPE A LINE AT STALL SPEED
011534  005367  000004                  DEC     CNT
011540  001373                          BNE     CT32B
011542  104012                          CHAIN
011544  000000                  CNT:    0
                                        ;NOTE:  FOR PRINT RATE TEST
                                        ;1)  TIME FIRST 20. LINES = APPROX 30 SECONDS.
                                        ;2)  CLEAR STALL BIT (BIT14 IN BIS #BIT14,PRGID).
                                        ;    TIME ENTIRE TEST = APPROX 60 SECONDS.
```

```
                                ;PRG2=KEYBOARD TEST
011546  012767  011572  167562  PRG2:   MOV     #ET0,KSTART
011554  052767  000200  167570          BIS     #BIT7,PRGID
011562  104000                          TYPE
011564  013776                          KMSG1
011566  000167  170620                  JMP     $RSET
                                ;TEST THAT PRESSING KEY SETS DONE FLAG;
011572  000000          ET0:    0                               ;TEST #
011574  011676                          ET1                     ;NEXT TEST;
011576  012767  000005  167662          MOV     #5,CTRA
011604  104006          ET0A:   $TRDRV
011606  011642                          ET0B
011610  104000                          TYPE                    ;TYPE "PRESS A KEY WITHIN 10 SECS;"
011612  014014                          KMSG2
011614  052777  000100  167470          BIS     #BIT6,@TKS      ;ENABLE KYBD INTERRUPT;
011622  005067  166150                  CLR     PSW
011626  104400                          DELAY                   ;WAIT 10 SECONDS
011630  023420                          10000.
011632  104000                          TYPE                    ;TYPE "NO KEYBOARD REQUEST;"
011634  014216                          KMSG6
011636  104010                          EHALT                   ;HALT;
011640  000411                  BR      ET0CA
011642  105777  167444          ET0B:   TSTB    @TKS            ;TEST FOR DONE BIT ON
011646  100403                  BMI     ET0C                    ;BRANCH IF DONE BIT SET;
011650  104000                          TYPE                    ;DONE BIT NOT SET; TYPE(FALSE  KEY=
011652  014244                          KMSG7                   ;BOARD OR READER INTERRUPT;
011654  104010                          EHALT                   ;HALT
011656  012716  011664          ET0C:   MOV     #ET0CA,@X6
011662  000002                  RTI                             ;EXIT INTERRUPT;
011664  104011          ET0CA:  $RESET
011666  005367  167774                  DEC     CTRA            ;DONE 5 TIMES?
011672  001344                  BNE     ET0A                    ;BRANCH IF NOT DONE;
011674  104012                  CHAIN                           ;CHAIN
                                ;ECHO TEST; KEYED CHARACTER IS TYPED; RUBOUT ENDS ROUTINE;
011676  000001          ET1:    1                               ;TEST #
011700  011756                  ET2                             ;NEXT TEST;
011702  104000                  TYPE                            ;TYPE TITLE AND INSTRUCTIONS,
011704  014054                  KMSG3
011706  105777  167400          ET1A:   TSTB    @TKS            ;WAIT FOR DONE FLAG
011712  100375                  BPL     .-4
011714  117767  167374  167532          MOVB    @TKB,CRBUF      ;MOVE KYBD CHAR TO CRBUF;
011722  116777  167526  167370          MOVB    CRBUF,@TPB      ;ECHO CHAR READ;
011730  105777  167362                  TSTB    @TPS            ;WAIT FOR PRINTER DONE;
011734  100375                  BPL     .-4
011736  042767  000200  167510          BIC     #BIT7,CRBUF     ;CLEAR BIT 7 FROM CRBUF;
011744  122767  000177  167502          CMPB    #177,CRBUF      ;COMPARE CRBUF TO RUBOUT (177)
011752  001355                  BNE     ET1A                    ;BRANCH IF NOT RUBOUT (177)
011754  104012                  CHAIN                           ;CHAIN
```

```
                                ;OCTAL EQUIVALENT TEST; THE OCTAL EQUIVALENT OF ANY CHARACTER KEYED
                                ;IS PRINTED, RUBOUT ENDS ROUTINE;.
011756  000002                ET21    2                               ;TEST #
011760  177777                        -1                              ;LAST TEST
011762  104001                        TYPES                           ;TYPE TITLE AND INSTRUCTIONS.
011764  014156                        KMSG4
011766  014067                        KMSG3A
011770  177777                        -1
011772  005067  167456                CLR     CRBUF
011776  105777  167310        ET2A1   TSTB    @TKS                    ;WAIT FOR DONE FLAG.
012002  100375                        BPL     .-4
012004  117767  167304  167442        MOVB    @TKB,CRBUF              ;CHARACTER TO CRBUF
012012  004567  173276                JSR     %5,ACNV4                ;CONVERT CHAR IN CRBUF TO
012016  001454                        CRBUF                           ;PRINTABLE OCTAL
012020  014210                        OCTEQV
012022  104000                        TYPE                            ;TYPE OCTAL EQUIVALENT
012024  014206                        KMSG5
012026  042767  000200  167420        BIC     #BIT7,CRBUF             ;CLEAR BIT 7 FROM CRBUF
012034  022767  000177  167412        CMP     #177,CRBUF              ;TEST FOR RUBOUT CHARACTER;
012042  001355                        BNE     ET2A                    ;BRANCH IF NOT RUBOUT (177).
012044  104012                        CHAIN                           ;CHAIN;
```

```
                                        ;PRG3-PRINTER EXERCISER, KEYBOARD CONTROLLED,
                                        ;TYPES LINES WITH ANY 5 CHARACTERS', STALLS OR FULL SPEED',
012046  004767  173666          PRG3:   JSR     X7,STBF         ;SET UP BUFFER',
012052  104000                          TYPE                    ;TYPE TITLE
012054  014272                          P7MG1
012056  052767  040000  167266  HTA:    BIS     #BIT14,PRGID    ;SET STALL BIT IN PRGID,
012064  012767  177600  172546          MOV     #177600,STLMSK  ;SET STALL MASK
012072  012703  014557                  MOV     #BLOCK1,X3
012076  104000                          TYPE                    ;TYPE "TYPE IN DATA",
012100  014320                          P7MG2
012102  012767  000006  167356          MOV     #6,CTRA         ;CHAR COUNT TO CTRA,
012110  004767  000060          HTB:    JSR     X7,GKBCR        ;GET AND STORE KYBD CHARACTER',
012114  005367  167346                  DEC     CTRA            ;GOT 6 CHARACTERS?
012120  001373                          BNE     HTB             ;BRANCH IF NOT 6 CHARS YET',
012122  042767  000200  167324          BIC     #BIT7,CRBUF
012130  122767  000177  167316          CMPB    #177,CRBUF      ;CHECK 6TH CHAR FOR RUBOUT',
012136  001003                          BNE     HTC             ;BRANCH IF NOT A RUBOUT,
012140  042767  040000  167204          BIC     #BIT14,PRGID    ;RUBOUT, CLEAR STALL BIT IN PRGID,
012146  004567  173254          HTC:    JSR     X5,BMOVE        ;FILL 80 CHAR LINE',
012152  014557                          BLOCK1
012154  014564                          BLOCK1+5
012156  000120                          80.
012160  004767  173434          HTD:    JSR     X7,TYPLN        ;TYPE LINE',
012164  005767  165400                  TST     SR              ;CHANGE DATA? (SR15=1)',
012170  100732                          BMI     HTA             ;YES', GO CHANGE DATA
012172  000772                          BR      HTD             ;NO CONTINUE WITH SAME DATA',
012174  105777  167112          GKBCR:  TSTB    @TKS            ;WAIT FOR DONE FLAG',
012200  100375                          BPL     .-4
012202  117767  167106  167244          MOVB    @TKB,CRBUF      ;CHARACTER TO CRBUF',
012210  116723  167240                  MOVB    CRBUF,(3)+      ;CHARACTER TO LINE BUFFER',
012214  116700  167234                  MOVB    CRBUF,X0
012220  004767  173240                  JSR     X7,LBPCH        ;ECHO CHARACTER,
012224  000207                          RTS     X7
```

```
                                ;
                                ;
                                ;PRG4=PUNCH CLOCK ADJUSTMENT ROUTINE;
                                ;OUTPUTS CHARACTER SET IN LEFT HALF OF SR, AND
                                ;STALLS FOR NUMBER OF MILLISECONDS SET IN RIGHT HALF OF SR;
012226  104005                  PRG4:   CHALT                   ;HALT TO SET SR,
012230  004767  000036          :TA:    JSR     X7,C1112        ;GO OUTPUT CHARACTER SET IN LEFT
012234  000775                          BR      :TA             ;HALF OF SR AND STALL PER SR RIGHT;
                                ;
                                ;
                                ;PRG5=READER CLOCK ADJUSTMENT ROUTINE;
                                ;PERFORMS SAME FUNCTION AS PRG11, AND IN ADDITION,
                                ;USING THE PUNCH MAINTENANCE BIT, SHIFTS OUTPUT OF PUNCH
                                ;SHIFT REGISTER ONTO THE READER BUFFER; THE CONTENTS OF THE
                                ;READER BUFFER ARE THEN "FIXED" ON THE CONSOLE DATA LIGHTS
                                ;BY ISSUING A RESET WITH CONTENTS OF READER BUFFER LOADED IN R0,
012236  104005                  PRG5:   CHALT                   ;HALT TO SET SR,
012240  004767  000020          JTA:    JSR     X7,C1112M       ;GO OUTPUT CHARACTER FROM SR LEFT AND
012244  017700  167044                  MOV     @TKB,X0         ;STALL PER SR RIGHT, (TKB) TO R0,
012250  000005                          RESET                   ;MPIXM (TKB) IN DATA LIGHTS,
012252  000005                          RESET
012254  000005                          RESET
012256  000005                          RESET
012260  000005                          RESET
012262  000766                          BR      JTA             ;REPEAT;
                                ;
012264  052777  000004  167024  C1112M: BIS     #4,@TPS         ;SET MAINTENANCE MODE (PUNCH);
012272  116767  165272  000022  C1112:  MOVB    SR,XTY          ;STALL COUNT TO XTY;
012300  005767  000016                  TST     XTY             ;DISREGARD 0 DELAY;
012304  001002                          BNE     C1112A
012306  005267  000010                  INC     XTY
012312  116777  165253  167000  C1112A: MOVB    SR+1,@TPB       ;LOAD PUNCH BUFFER;
012320  104400                          DELAY                   ;DELAY (APPROXIMATELY) THE NUMBER OF
012322  000000                  XTY:    OPEN                    ;MSECS; SPECIFIED AT SR RIGHT
012324  000207                          RTS     X7              ;EXIT
```

```
                                ;PRG6-MAINTENANCE MODE SINGLE CHARACTER DATA TEST;
                                ;WITH MAINTENANCE MODE SET, OUTPUTS ONTO PUNCH BUFFER AND BACK ONTO
                                ;READER BUFFER THE CHARACTER SET IN SR LEFT, THE CHARACTER IN THE
                                ;READER BUFFER IS COMPARED TO THE CHARACTER IN SR LEFT; IF THE 2 CHARACTERS
                                ;DISAGREE THE PROGRAM HALTS, THE DATA LIGHTS WILL THEN CONTAIN:
                                ;
                                ;LEFT HALF: THE EXPECTED CHARACTER (SR LEFT);
                                ;RIGHT HALF: THE CHARACTER IN THE READER BUFFER;
012326  104005                  PRG6:   CHALT                   ;HALT TO SET SR;
012330  052777  000004  166760  KTA:    BIS     #4,@TPS         ;SET MAINTENANCE MODE;
012336  105777  166754          KTB:    TSTB    @TPS            ;WAIT FOR READY;
012342  100375                          BPL     .-4
012344  116767  165221  167103          MOVB    SR+1,CRBUF+1    ;S/B CHAR TO CRBUF+1;
012352  116777  167077  166740          MOVB    CRBUF+1,@TPB    ;OUTPUT CHARACTER;
012360  105777  166726                  TSTB    @TKS            ;WAIT FOR READER DONE FLAG;
012364  100375                          BPL     .-4
012366  117767  166722  167060          MOVB    @TKB,CRBUF      ;CHAR READ TO CRBUF;
012374  104004                          DATCHK                  ;GO CHECK AGAINST S/B CHAR;
012376  000754                          BR      KTA             ;REPEAT;

                                ;PRG7-MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN DATA TEST;
                                ;PERFORMS SAME OPERATION AS PRG6, EXCEPT THAT SPECIAL BINARY COUNT
                                ;PATTERN IS USED;
012400  004767  172510  PRG7:   JSR     %7,INBIN        ;INITIALIZE BINARY COUNT
012404  012767  177000  172226          MOV     #177000,STLMSK  ;SET STALL LIMIT
012412  052767  040000  166732          BIS     #BIT14,PRGID    ;ALLOW STALLS
012420  052777  000004  166670  LTA:    BIS     #4,@TPS         ;SET MAINTENANCE MODE;
012426  032767  000400  165134          BIT     #BIT8,SR        ;CHECK STALL SWITCH
012434  001001                          BNE     LTB             ;BRANCH IF NO STALL WANTED
012436  104002                          STALL                   ;STALL
012440  105777  166652  LTB:    TSTB    @TPS            ;WAIT FOR READY;
012444  100375                          BPL     .-4
012446  004767  172546                  JSR     %7,GTBINP       ;GET BIN CHARACTER;
012452  116167  166777                  MOVB    %1,CRBUF+1      ;MOVE TO S/B CHAR;
012456  116177  166636                  MOVB    %1,@TPB         ;OUTPUT BIN CHARACTER;
012462  105777  166624                  TSTB    @TKS            ;WAIT FOR READER DONE;
012466  100375                          BPL     .-4
012470  117767  166620  166756          MOVB    @TKB,CRBUF      ;CHAR IN READ BUFFER TO CRBUF;
012476  104004                          DATCHK                  ;GO CHECK AGAINST S/B CHAR;
012500  005767  165336                  TST     MONITR          ;MONITOR LOAD?
012504  001401                          BEQ     LTC             ;NO=CONTINUE
012506  000207                          RTS     %7              ;YES=RETURN
012510  000743  LTC:    BR      LTA             ;CONTINUE;


                                ;PRG10, NON-PRINTING CHAR TEST

012512  104000  PRG10:  TYPE                    ;PRINT TEST TITLE
012514  014351                  NPCT
012516  012767  000120  000124          MOV     #80.,.CCNT      ;CHAR COUNT
012524  012700  000101  PRG10A: MOV     #'A,%0          ;PRINTABLE CHARACTER
012530  004767  172730          JSR     %7,LSPCH        ;PRINT PRINTABLE CHAR

                                ;WHY ISN'T 140 THE FIRST NON-PRINTING CHARACTER?
```

```
012534  012767  000177  177002          MOV     #177,CNT        ;FIRST NON-PRINTABLE CHAR
012542  016700  176776          PRG10B: MOV     CNT,R0
012546  004767  172712                  JSR     R7,LSPCH        ;PRINT NON-PRINTABLE CHAR
012552  105267  176766                  INCB    CNT
012556  022767  000012  176760          CMP     #12,CNT         ;LINE FEED
012564  001366                          BNE     PRG10B          ;NO
012566  012700  000013                  MOV     #13,R0          ;PRINT NON-PRINTABLES
012572  004767  172666                  JSR     R7,LSPCH
012576  012700  000014                  MOV     #14,R0
012602  004767  172656                  JSR     R7,LSPCH
012606  012767  000016  176730          MOV     #16,CNT
012614  016700  176724          PRG10C: MOV     CNT,R0          ;PRINT NEST SET OF NON-PRINTABLES
```

```
012620  004767  172640              JSR     %7,LSPCH
012624  005267  176714              INC     CNT
012630  022767  000040  176706      CMP     #40,CNT
012636  001366                      BNE     PRG10C
012640  005367  000004              DEC     CCNT            ;DEC CHAR COUNT
012644  001327                      BNE     PRG10A          ;BRANCH IF NOT END OF TEST
012646  000721                      BR      PRG1B           ;RERUN TEST
012650  000000              CCNT:   0                       ;CHAR COUNT

                            ;PRG12, WORST CASE NOISE
                            ;THIS PROGRAM READ IN DATA FROM THE KEYBOARD UNTIL (1) A
                            ;RUBOUT CHARACTER IS SENSED OR (2) UNTIL AN EIGHTY CHARACTER
                            ;BUFFER IS FILLED.  IT THEN ECHOS THE DATA UNTIL SWITCH 15 IS PUT UP.

012652  104000              PRG11:  TYPE                    ;TYPE PROGRAM TITLE
012654  014411                      WSTCN
012656  104000              PRG11C: TYPE                    ;TYPE
012660  014436                      RUB80                   ;RUBOUT OR 80 CHAR ENDS DATA
012662  012767  000120  177760      MOV     #80.,,CCNT      ;SET CHAR COUNT = 80
012670  012701  014557              MOV     #DEND+2,R1
012674  105777  166412      PRG11A: TSTB    @TKS            ;TEST FOR KEYBOARD DONE
012700  100375                      BPL     .-4
012702  117711  164406              MOVB    @TKB,@R1        ;LOAD CHAR INTO BUFFER
012706  111100                      MOVB    @R1,%0          ;ECHO CHAR
012710  004767  172550              JSR     %7,LSPCH
012714  122127  000177              CMPB    (R1)+,#177      ;TEST FOR RUBOUT
012720  001403                      BEQ     PRG11B          ;DUMP BUFFER IF END OF DATA
012722  005367  177722              DEC     CCNT            ;DECREASE CHAR COUNT
012726  001362                      BNE     PRG11A          ;BRANCH IF NOT 80 CHAR
012730  004767  000010      PRG11B: JSR     %7,DUMP         ;DUMP BUFFER
012734  005767  164430              TST     $R              ;TEST FOR END ROUTINE
012740  100373                      BPL     PRG11B          ;CONTINUE DUMP IF NOT END
012742  000745                      BR      PRG11C          ;ASK FOR NEW DATA

012744  012700  000015      DUMP:   MOV     #15,%0          ;CARRIAGE RETURN
012750  004767  172510              JSR     %7,LSPCH
012754  012700  000012              MOV     #12,%0          ;LINE FEED
012760  004767  172500              JSR     %7,LSPCH
012764  012702  014557              MOV     #DEND+2,R2      ;R2 = BEGINNING OF BUFFER
012770  112200      DMPA:   MOVB    (R2)+,%0        ;FETCH CHARACTER FROM BUFFER
012772  004767  172466              JSR     %7,LSPCH        ;PRINT CHAR
012776  020201                      CMP     R2,R1           ;TEST FOR END OF BUFFER
013000  001373                      BNE     DMPA            ;BRANCH IF NOT END
013002  000207                      RTS     %7              ;END OF BUFFER

                            ;PROGRAM 13, LAST CHARACTER VISIBILITY

013004  104000              PRG12:  TYPE                    ;TYPE TITLE OF TEST
013006  014512                      LCVIS
013010  012767  000120  177632      MOV     #80.,,CCNT      ;SET CHAR COUNT = 80.
013016  012700  000015              MOV     #15,%0          ;CR
013022  004767  172436              JSR     %7,LSPCH        ;
013026  012700  000012              MOV     #12,%0          ;LF
013032  004767  172426              JSR     %7,LSPCH
```

```
013036  012700  000102          PRG12A: MOV     #'B,%0          ;LOAD TEST CHAR
013042  004767  172416                  JSR     %7,LSPCH        ;PRINT
013046  104400                          DELAY
013050  003720                          2000.                   ;DELAY 2 SECONDS FOR
                                                                ;PRINT HEAD TO STEP ASIDE
013052  005367  177572                  DEC     CCNT            ;DEC CHAR COUNT
013056  001367                          BNE     PRG12A          ;BRANCH IF NOT 80 CHAR
013060  000751                          BR      PRG12           ;CONTINUE TEST
013062  004777  164754          LOGICAL:        JSR     %7,@MONITR
013066  000240                          NOP
013070  000240                          NOP
013072  000240                          NOP
```

```
013074  045                     DONE1   ,ASCII  /%# TEST COMPLETED#/
013075  043
013076  040
013077  124
013100  105
013101  123
013102  124
013103  040
013104  103
013105  117
013106  115
013107  120
013110  114
013111  105
013112  124
013113  105
013114  104
013115  100

013116  045                     ERRORM1 ,ASCII  /%# SWITCH OR CONSOLE ERROR DEVICE /
013117  043
013120  040
013121  123
013122  127
013123  111
013124  124
013125  103
013126  110
013127  040
013130  117
013131  122
013132  040
013133  103
013134  117
013135  116
013136  123
013137  117
013140  114
013141  105
013142  040
013143  105
013144  122
013145  122
013146  117
013147  122
013150  040
013151  104
013152  105
013153  126
013154  111
013155  103
013156  105
013157  040

013160  041                     ERRORA1 ,ASCII  /176500 PROGRAM ADDRESS /
```

```
013161      067
013162      066
013163      065
013164      060
013165      060
013166      040
013167      128
013170      122
013171      117
013172      107
013173      122
013174      101
013175      115
013176      040
013177      101
013200      104
013201      104
013202      122
013203      105
013204      123
013205      123
013206      040

013207      060                 ERRORB: .ASCII  '0000000 '
013210      060
013211      060
013212      060
013213      060
013214      060
013215      100
013216      040

013217      045                 CERR:   .ASCII  '%# ERROR DEVICE '
013220      043
013221      040
013222      105
013223      122
013224      122
013225      117
013226      122
013227      040
013230      104
013231      105
013232      124
013233      111
013234      103
013235      105
013236      040

013237      061                 CERRA:  .ASCII  '176500 PROGRAM ADDRESS '
013240      067
013241      066
013242      065
013243      060
013244      060
```

```
013245      040
013246      120
013247      122
013250      117
013251      127
013252      122
013253      101
013254      115
013255      040
013256      101
013257      124
013260      104
013261      122
013262      105
013263      103
013264      040

013265      040                 CERRB1  ,ASCII  '0000000'
013266      060
013267      060
013270      060
013271      060
013272      060
013273      060
013274      060

013275      045                 DERR1   ,ASCII  'X# DATA ERROR DEVICE '
013276      043
013277      040
013300      104
013301      101
013302      124
013303      101
013304      040
013305      105
013306      122
013307      122
013310      117
013311      122
013312      040
013313      104
013314      105
013315      126
013316      111
013317      103
013320      105
013321      040

013322      061                 DERRA1  ,ASCII  '176500'
013323      067
013324      066
013325      065
013326      060
013327      060
013330      100
```

```
013331  045                 ABORT:  ;ASCII  'X# SWITCH ERROR TEST ABORTED#/
013332  043
013333  040
013334  123
013335  127
013336  111
013337  124
013340  103
013341  110
013342  040
013343  105
013344  122
013345  122
013346  117
013347  122
013350  040
013351  124
013352  105
013353  123
013354  124
013355  040
013356  101
013357  102
013360  117
013361  122
013362  124
013363  105
013364  104
013365  100

013366  047                 A33WP6: ;BYTE   047,137,127,057,127,137
013367  137
013370  127
013371  057
013372  127
013373  137
013374  047                 A35WP6: ;BYTE   047,133,077,103,077,133
013375  133
013376  077
013377  103
013400  077
013401  133
013402  101                 A:      ;BYTE   101,102,103
013403  102
013404  103
013405  104                 D:      ;BYTE   104,105,106
013406  105
013407  106
013410  107                 G:      ;BYTE   107,110,111
013411  110
013412  111
013413  112                 J:      ;BYTE   112,113,114
013414  113
```

```
013415      114
013416      115                 M1      ,BYTE   115,116,117
013417      116
013420      117
013421      120                 P1      ,BYTE   120,121,122
013422      121
013423      122
013424      123                 S1      ,BYTE   123,124,125
013425      124
013426      125
013427      126                 V1      ,BYTE   126,127,130
013430      127
013431      130
013432      131                 Y1      ,BYTE   131,132,060
013433      132
013434      133
013435      061                 ONE1    ,BYTE   061,062,063
013436      062
013437      063
013440      064                 FOUR1   ,BYTE   064,065,066
013441      065
013442      066
013443      067                 SEVEN1  ,BYTE   067,070,071
013444      070
013445      071
013446      041                 C41     ,BYTE   041,042,043
013447      042
013450      043
013451      044                 C44     ,BYTE   044,045,046
013452      045
013453      046
013454      047                 C47     ,BYTE   047,050,051
013455      050
013456      051
013457      052                 C52     ,BYTE   052,053,054
013460      053
013461      054
013462      055                 C55     ,BYTE   055,056,057
013463      056
013464      057
013465      072                 C72     ,BYTE   072,073,074
013466      073
013467      074
013470      075                 C75     ,BYTE   075,076,077
013471      076
013472      077
013473      100                 C100    ,BYTE   100,133,134
013474      133
013475      134
013476      135                 C135    ,BYTE   135,136,137
013477      136
013500      137
013501      377                 C377    ,BYTE   377,000,377
013502      000
013503      377
```

```
013504      045                 TBTST:  .ASCII  'X#SPACE TESTX#'
013505      043
013506      123
013507      120
013510      101
013511      103
013512      105
013513      040
013514      124
013515      105
013516      123
013517      124
013520      045
013521      043

013522      040                 TBMRK:  .ASCII  '        /@'
013523      040
013524      040
013525      040
013526      040
013527      040
013530      040
013531      040
013532      057
013533      100

013534      040                 TBMRK1: .ASCII  '        /@'
013535      040
013536      040
013537      040
013540      040
013541      040
013542      040
013543      057
013544      100

013545      045                 CRLF:   .ASCII  'X@'
013546      100

013547      055                 RM33A:  .ASCII  '----!@'
013550      055
013551      055
013552      055
013553      111
013554      100

013555      055                 RM33B:  .ASCII  '-!-@'
013556      111
013557      055
013560      100

013561      055                 RM37A:  .ASCII  '-----!-!@'
013562      055
013563      055
013564      055
```

```
013565  111
013566  055
013567  111
013570  100

013571  134                 SPTSTC: .ASCII  '\ '
013572  040
013573  100

013574  045                 CRTST:  .ASCII  'X-CARRIAGE RETURN TESTX-'
013575  043
013576  103
013577  101
013600  122
013601  122
013602  111
013603  101
013604  107
013605  105
013606  040
013607  122
013610  105
013611  124
013612  125
013613  122
013614  116
013615  040
013616  124
013617  105
013620  123
013621  124
013622  045
013623  043
013624  100

013625  045                 RMTST:  .ASCII  'X-RIGHT MARGIN TESTX-'
013626  043
013627  122
013630  111
013631  107
013632  110
013633  124
013634  040
013635  115
013636  101
013637  122
013640  107
013641  111
013642  116
013643  040
013644  124
013645  105
013646  123
013647  124
013650  045
```

```
013691      043
013692      100

013693      045                 SPTST:  .ASCII  '%#SPACE  TEST%#@'
013694      043
013695      123
013696      120
013697      101
013700      103
013701      105
013702      040
013703      040
013704      124
013705      105
013706      123
013707      124
013710      045
013711      043
013712      100

013713      045                 LFTST:  .ASCII  '%#LINE FEED TEST%#@'
013714      043
013715      114
013716      111
013717      116
013720      105
013721      040
013722      106
013723      105
013724      105
013725      104
013726      040
013727      124
013730      105
013731      123
013732      124
013733      045
013734      043
013735      100

013736      045                 CHRTST: .ASCII  '%#CHARACTER TEST$%#@'
013737      043
013740      103
013741      110
013742      101
013743      122
013744      101
013745      103
013746      124
013747      105
013750      122
013751      040
013752      124
013753      105
013754      123
```

```
013735      124
013736      123
013737      045
013740      043
013741      100

013742      045                 WCPTST: .ASCII  /X#WORST CASE PATTERN TEST X#0/
013743      043
013744      127
013745      117
013746      122
013747      123
013750      124
013751      040
013752      043
013753      101
013754      123
013755      105
013756      040
013757      120
013760      101
013761      124
013762      124
013763      105
013764      122
013765      116
013766      040
013767      124
013770      105
013771      123
013772      124
013773      045
013774      043
013775      100

013776      045                 KMSG1:  .ASCII  /X#KYBD TEST X#0/
013777      043
014000      113
014001      131
014002      102
014003      104
014004      040
014005      124
014006      105
014007      123
014010      124
014011      045
014012      043
014013      100

014014      045                 KMSG2:  .ASCII  /XPRESS A KEY WITHIN 10 SECONDS,0/
014015      120
014016      122
014017      105
014020      123
```

```
014021      123
014022      040
014023      101
014024      040
014025      113
014026      105
014027      131
014030      040
014031      127
014032      111
014033      124
014034      110
014035      111
014036      116
014037      040
014040      061
014041      060
014042      040
014043      123
014044      105
014045      103
014046      117
014047      116
014050      104
014051      123
014052      056
014053      100

014054      045                 KMSG31  ,ASCII  '%ECHO TEST'
014055      103
014056      105
014057      103
014060      110
014061      117
014062      040
014063      124
014064      105
014065      123
014066      124

014067      045                 KMSG3A1 ,ASCII  '%CHARACTER KEYED WILL BE TYPED,'
014070      103
014071      110
014072      101
014073      122
014074      101
014075      103
014076      124
014077      105
014100      122
014101      040
014102      113
014103      105
014104      131
014105      105
```

```
014106  104
014107  040
014110  127
014111  111
014112  114
014113  114
014114  040
014115  062
014116  065
014117  040
014120  124
014121  110
014122  101
014123  124
014124  124
014125  040
014126  105
014127  116
014130  104
014131  123
014132  040
014133  045
014134  043
014135  100

014136  045                    .ASCII  '%RUBOUT ENDS ROUTINE',%#0/
014137  122
014140  125
014141  102
014142  117
014143  125
014144  124
014145  040
014146  105
014147  116
014150  104
014151  123
014152  040
014153  122
014154  117
014155  125
014156  124
014157  111
014160  116
014161  105
014162  056
014163  045
014164  043
014165  100

014166  045                KMSG41  .ASCII  '%#OCTAL EQUIVALENT TEST#/
014167  043
014170  117
014171  103
014172  124
014173  101
014174  114
014175  040
014176  105
014177  121
014200  125
014201  111
014202  126
```

```
014173      101
014174      114
014175      105
014176      116
014177      124
014200      040
014201      124
014202      105
014203      123
014204      124
014205      120

014206      045                 KMSG5:  ,ASCII  '% '
014207      040

014210      040                 OCTEQV: ,ASCII  '    %0'
014211      040
014212      040
014213      040
014214      045
014215      100

014216      045                 KMSG6:  ,ASCII  '%NO KEYBOARD REQUEST',0
014217      116
014220      117
014221      040
014222      113
014223      105
014224      131
014225      102
014226      117
014227      101
014230      122
014231      104
014232      040
014233      122
014234      105
014235      121
014236      125
014237      105
014240      123
014241      124
014242      000
014243      100

014244      045                 KMSG7:  ,ASCII  '%FALSE KYBD INTERRUPT0'
014245      106
014246      101
014247      114
014250      123
014251      105
014252      040
014253      113
014254      131
014255      102
```

```
014256      104
014257      040
014260      111
014261      116
014262      124
014263      135
014264      122
014265      122
014266      125
014267      120
014270      124
014271      120

014272      045                 P7MG1I  ,ASCII  /X-DISPLAY EXERCISER X-0/
014273      043
014274      104
014275      111
014276      123
014277      120
014300      114
014301      101
014302      131
014303      040
014304      105
014305      130
014306      105
014307      122
014310      103
014311      111
014312      123
014313      105
014314      122
014315      045
014316      043
014317      100

014320      045                 P7MG2I  ,ASCII  /X-TYPE IN DATA )0/
014321      043
014322      124
014323      131
014324      120
014325      105
014326      040
014327      111
014330      116
014331      040
014332      104
014333      101
014334      124
014335      101
014336      040
014337      072
014340      120

014341      125                 BKSUI   ,ASCII  /U 0/
```

```
014342      040
014343      120

014344      040                 DECVAL: .ASCII  '      '
014345      040
014346      040
014347      040
014350      040

014351      045                 NPCT:   .ASCII  'X#NON-PRINTING CHARACTER TESTX#0'
014352      043
014353      116
014354      117
014355      116
014356      055
014357      120
014360      122
014361      111
014362      116
014363      124
014364      111
014365      116
014366      107
014367      040
014370      103
014371      110
014372      101
014373      122
014374      101
014375      103
014376      124
014377      105
014400      122
014401      040
014402      124
014403      105
014404      123
014405      124
014406      045
014407      043
014410      100

014411      045                 WSTCN:  .ASCII  'X#WORST CASE NOISEX#0'
014412      043
014413      127
014414      117
014415      122
014416      123
014417      124
014420      040
014421      103
014422      101
014423      123
014424      105
014425      040
```

```
014426  116
014427  117
014430  111
014431  123
014432  105
014433  005
014434  043
014435  100

014436  045                 RUB001: .ASCII  /%%TYPE DATA - RUBOUT OR 80 CHAR ENDS DATA%%@/
014437  003
014440  124
014441  111
014442  120
014443  105
014444  040
014445  104
014446  101
014447  124
014450  101
014451  040
014452  125
014453  116
014454  102
014455  125
014456  124
014457  117
014460  040
014461  124
014462  040
014463  117
014464  122
014465  040
014466  070
014467  060
014470  040
014471  103
014472  110
014473  101
014474  122
014475  040
014476  105
014477  116
014500  104
014501  123
014502  040
014503  104
014504  101
014505  124
014506  101
014507  045
014510  043
014511  100

014512  045                 LCVIS1: .ASCII  /%%LAST CHARACTER VISIBILITY TEST%%@/
```

```
014513      043
014514      114
014515      171
014516      123
014517      124
014520      040
014521      103
014522      110
014523      101
014524      122
014525      101
014526      103
014527      124
014530      105
014531      122
014532      040
014533      126
014534      111
014535      123
014536      111
014537      102
014540      111
014541      114
014542      111
014543      124
014544      131
014545      040
014546      124
014547      105
014550      123
014551      124
014552      040
014553      043
014554      100

            000001                  DEND:   .END
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | 013402 | AT2 | 006160 | AT33B | 007530 | BDCNV | 005500 |
| A1ST | 005342 | AT20 | 006664 | AT33E | 007526 | BDCNVA | 005520 |
| A33WP6 | 013366 | AT20A | 006700 | AT34 | 007534 | BELL | 000007 |
| A35WP6 | 013374 | AT20B | 006710 | AT34A | 007550 | BIT0 | 000020 |
| ABORT | 013331 | AT20C | 006734 | AT34B | 007572 | BIT1 | 000002 |
| ACNV | 005352 | AT20D | 006736 | AT34C | 007576 | BIT10 | 002000 |
| ACNVA | 005364 | AT20E | 006730 | AT34E | 007570 | BIT11 | 004000 |
| ACNVB | 005366 | AT21 | 006742 | AT35 | 007602 | BIT12 | 010000 |
| ACNVE | 005306 | AT21A | 006756 | AT35A | 007616 | BIT13 | 020000 |
| ACNVC | 005334 | AT21B | 006766 | AT35B | 007640 | BIT14 | 040000 |
| ACNVE | 002110 | AT21C | 007010 | AT35E | 007650 | BIT15 | 100000 |
| ACNVM | 005336 | AT21E | 007020 | AT36 | 007656 | BIT2 | 000004 |
| ACNVW | 005330 | AT22 | 007026 | AT36A | 007672 | BIT3 | 000010 |
| ADTENP | 005006 | AT22A | 007052 | AT36B | 007726 | BIT4 | 000020 |
| ARDA | 003770 | AT22B | 007106 | AT36C | 007730 | BIT5 | 000040 |
| ARDB | 004004 | AT22C | 007110 | AT37 | 007740 | BIT6 | 000100 |
| AREAD | 003760 | AT22E | 007102 | AT37A | 007750 | BIT7 | 000200 |
| AT0 | 006074 | AT23 | 007120 | AT37B | 007776 | BIT8 | 000400 |
| AT0A | 006112 | AT23A | 007140 | AT37C | 010006 | BIT9 | 001000 |
| AT0B | 006116 | AT23B | 007162 | AT37D | 010022 | BKSU | 014341 |
| AT0E | 006122 | AT23C | 007172 | AT37E1 | 007774 | BLK2 | 014702 |
| AT1 | 006126 | AT23D | 007206 | AT37E2 | 010026 | BLKBB | 014700 |
| AT11 | 006374 | AT23E1 | 007160 | AT3A | 006230 | BLKCC | 015012 |
| AT11A | 006404 | AT23E2 | 007212 | AT3B | 006234 | BLOCK1 | 014597 |
| AT11B | 006414 | AT24 | 007220 | AT3E | 006240 | BLOCK2 | 014671 |
| AT12 | 006420 | AT24A | 007236 | AT4 | 006244 | BLOCKA | 014555 |
| AT12A | 006430 | AT24B | 007260 | AT4B | 010034 | BLOCKB | 014667 |
| AT12B | 006434 | AT24C | 007300 | AT4BA | 010050 | BLOCKC | 015001 |
| AT12C | 006446 | AT24E1 | 007254 | AT4BB | 010106 | BMOVA | 005436 |
| AT12D | 006460 | AT24E2 | 007276 | AT4BC | 010110 | BMOVE | 005426 |
| AT12E | 006464 | AT25 | 007304 | AT41 | 010120 | BRCTR | 004264 |
| AT12F | 006466 | AT25A | 007322 | AT41A | 010140 | BREAD | 004166 |
| AT13 | 006472 | AT25B | 007344 | AT41B | 010164 | BREADA | 004224 |
| AT13A | 006502 | AT25E | 007342 | AT41C | 010176 | BREADB | 004234 |
| AT13B | 006506 | AT2A | 006176 | AT42 | 010206 | BREADC | 004254 |
| AT13C | 006524 | AT2B | 006202 | AT42A | 010222 | BSYNC | 004726 |
| AT13E | 006522 | AT2E | 006206 | AT42B | 010250 | BYPMAN | 000400 |
| AT14 | 006530 | AT3 | 006212 | AT42E | 010246 | C100 | 013473 |
| AT14A | 006540 | AT30 | 007350 | AT43 | 010254 | C1112 | 012272 |
| AT14B | 006544 | AT30A | 007360 | AT43A | 010272 | C1112A | 012312 |
| AT14C | 006564 | AT30B | 007370 | AT43B | 010276 | C1112M | 012264 |
| AT14E | 006562 | AT30E | 007366 | AT4A | 006262 | C135 | 013476 |
| AT17 | 006570 | AT31 | 007374 | AT4B | 006304 | C377 | 013501 |
| AT17A | 006600 | AT31A | 007404 | AT4C | 006324 | C41 | 013446 |
| AT17B | 006604 | AT31B | 007426 | AT4E1 | 006300 | C44 | 013451 |
| AT17C | 006624 | AT31E | 007424 | AT4E2 | 006322 | C47 | 013454 |
| AT17D | 006652 | AT32 | 007432 | AT5 | 006330 | C52 | 013457 |
| AT17E | 006646 | AT32A | 007442 | AT5A | 006346 | C55 | 013462 |
| AT17F | 006660 | AT32B | 007470 | AT5B | 006370 | C72 | 013465 |
| AT1A | 006144 | AT32E | 007466 | AT5E | 006366 | C75 | 013470 |
| AT1B | 006150 | AT33 | 007474 | B | 002174 | CC | 177776 |
| AT1E | 006154 | AT33A | 007504 | BCHECK | 004666 | CCNT | 012650 |

```
CERR    013217      CT21    011324      DUMP    012744      ICTR    001346
CERRA   013237      CT22    011336      EHALT   104010      INBIN   005114
CERRB   013266      CT23    011350      EHLT    001520      INCPRG  002370
CERRW   002066      CT24    011362      EHLTA   001622      INCRTN  002586
CHAIN   104012      CT25    011374      EHLTB   001532      ITA     012230
CHAINN  002512      CT26    011406      EMTA    003062      J       013413
CHALT   104005      CT27    011420      EMTINT  003036      JTA     012240
CHK33   003234      CT2A    010524      EMTTAB  001402      KMSG1   013776
CHK330  003246      CT2B    010544      END     002212      KMSG2   014014
CHK35   003250      CT2C    010552      ERCTR   001464      KMSG3   014054
CHKASR  104022      CT2D    010570      ERR     001732      KMSG3A  014067
CHLT    001476      CT2E    010626      ERRA    001752      KMSG4   014156
CHLTA   001514      CT3     010636      ERRB    002060      KMSG5   014206
CHLTR   001510      CT30    011432      ERRH    001630      KMSG6   014216
CHNA    002546      CT31    011444      ERROR   104003      KMSG7   014244
CHNAA   002574      CT32    011456      ERRORA  013160      KSTART  001336
CHNB    002610      CT32A   011502      ERRORB  013207      KTA     012330
CHR1    001456      CT32B   011530      ERRORM  013116      KTB     012336
CHR2    001460      CT3A    010662      ET0     011572      LCVIS   014512
CHR3    001462      CT3B    010712      ET0A    011604      LPTST   013673
CHRTST  013716      CT4     010716      ET0B    011642      LOGICA  013062
CK33    104013      CT4A    010746      ET0C    011656      LOWADD  003602
CK35    104014      CT4B    010760      ET0CA   011664      LPRGSW  002000
CKASR   003266      CT5     011130      ET1     011676      LSPCH   005464
CLEAN   002776      CT6     011146      ET1A    011706      LSTADD  003604
CNT     011544      CT7     011160      ET2     011756      LTA     012420
CNVCTR  005600      CTRA    001466      ET2A    011776      LTB     012440
CONADD  001506      CTRB    001470      FBALL   006022      LTC     012510
CONIT   003312      CTRC    001472      FBFJ    005756      M       013416
CONVEC  001310      CTRD    001474      FBFJA   005766      MACHER  000004
CPRDY   005450      CURADD  003600      FBFJB   005774      MANUAL  100000
CPRDYA  005460      CURTST  001340      FORWD   002672      MONITR  000042
CRBUF   001454      CURVEC  003606      FORWDA  002724      NITRSW  004000
CRLF    013545      D       013405      FORWDB  002732      NOP     000240
CRTA    002374      DATCHK  104004      FOUR    013440      NPCT    014351
CRTST   013576      DATHLT  104017      G       013410      NPRTSW  020000
CTB     010340      DECVAL  014344      GETRDY  002414      NSTART  002274
CTBA    010364      DELAY   104000      GKBCR   012174      NTRCSW  010000
CTBB    010374      DEND    014555      GOTST   002750      NULL    004452
CTBC    010402      DERR    013275      GOTSTA  002770      NXTST   001344
CT1     010442      DERRA   013322      GRCNT   004642      OCTEQV  014210
CT10    011172      DERRW   001726      GTBIN   005152      ONE     013435
CT11    011204      DIGIT   005602      GTBINP  005220      OPEN    000000
CT12    011216      DLCNT   004600      GTRDYA  002426      OUT     004366
CT13    011230      DLY     004536      GTRDYB  002432      OUTCAR  004410
CT14    011242      DLYA    004556      GTRDYC  002450      OUTEX   004450
CT15    011254      DLYB    004562      GTRDYD  002476      OUTNUL  004402
CT16    011266      DMPA    012770      HLTSW   100000      OUTTYP  004372
CT17    011300      DONE    013074      HTA     012056      P       013421
CT1A    010466      DTCHK   001634      HTB     012110      P7MG1   014272
CT2     010506      DTCHKA  001730      HTC     012146      P7MG2   014320
CT20    011312      DTHLT   001722      HTD     012160      PFAIL   003126
```

```
PIND    005144          RNGEN   004114          SYCTRA  005110          XYY     012322
PDPSP   005726          RP1     004162          SYCTRB  005112          Y       013432
POPSP2  022626          RP2     004164          SYNCA   005006
PRG0    006062          RSTA    003424          SYNCB   005022
PRG1    010306          RSTART  003370          SYNCC   005054
PRG1@   012512          RSTB    003550          SYNCD   005060
PRG1@A  012524          RSTC    003574          SYNCE   005102
PRG1@B  012542          RSTPC   003704          TABP    011030
PRG1@C  012614          RSTPSW  003706          TABPA   011042
PRG11   012652          RSTREG  104021          TABPB   011052
PRG11A  012674          RSTRG   003600          TABPC   011070
PRG11B  012730          RTNNO   001342          TBCNT   011024
PRG11C  012656          RUB80   014436          TBMRK   013522
PRG12   013004          S       013424          TBMRK1  013534
PRG12A  013036          SAVR0   003162          TBTST   013504
PRG2    011346          SAVREG  104020          TCTR    005676
PRG3    012046          SAVRG   003610          TENPWR  005604
PRG4    012226          SCOPSW  040000          TKB     001314
PRG5    012236          SCOPTR  001350          TKLVL   001324
PRG6    012326          SEVEN   013443          TKS     001312
PRG7    012400          SHALT   002072          TKVTR   001322
PRGEND  002666          SHLTA   002106          TPB     001320
PRGID   001302          SPBOT   001304          TPBM    011000
PRGNUM  001334          SPCNT   011026          TPBMA   011010
PRGTAB  001354          SPTST   013653          TPLJA   005722
PRTY0   000000          SPTSTC  013571          TRLVL   001330
PRTY1   000040          SR      177570          TPS     001316
PRTY2   000100          SRESET  104011          TPVTR   001326
PRTY3   000140          SRSET   002412          TRPA    003214
PRTY4   000200          SRSETT  004076          TRPINT  003102
PRTY5   000240          SRTSW   001000          TRPTAB  001450
PRTY6   000300          STAL    004602          TTYTYP  001332
PRTY7   000340          STALA   004634          TYP     004266
PSW     177776          STALAA  004614          TYPA    004276
PT0     005140          STALB   004636          TYPC    004340
PT0P    005146          STALL   104002          TYPCLR  005650
PT1     005142          START   002134          TYPDAT  004502
PT1P    005150          STARTO  002330          TYPE    104000
R1      000001R         STBF    005740          TYPES   104001
R2      000002R         STLMSK  004640          TYPF    004454
RCMSK   004662          STLSPV  004046          TYPG    004466
RCNT    001452          STLSRV  004016          TYPLJ   005700
RESTAR  003164          STNUM   003576          TYPLA   005626
RETPVT  003710          STPCHV  104007          TYPLB   005632
RHLT    002104          STPPA   004064          TYPLN   005620
RIND    005136          STPRA   004034          TYPLNJ  104016
RM33A   013547          STRDRV  104006          TYPS    004504
RM33B   013555          SUBTEN  005540          TYPSA   004530
RM37A   013561          SUBTNA  005544          TYPSB   004532
RMB     011502          SUBTNB  005560          V       013427
RMTST   013625          SVRPC   003644          WCPTST  013742
RNCNT   004664          SVRPSW  003646          WSTCN   014411
```

```
ERRORS DETECTED: 0
RUN-TIME:  37 SECONDS
6K CORE USED
```